

# INFORMATYKA

# 4

KLASA

EWA GAŚIENICA-SAMEK  
ANDRZEJ GAŚIENICA-SAMEK



instaKod

PROGRAMOWANIE OD PODSTAW



Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych do kształcenia ogólnego do nauczania informatyki na podstawie opinii rzeczoznawców:

...

Etap edukacyjny: II

Typ szkoły: szkoła podstawowa

Rok dopuszczenia: 2019

**Numer ewidencyjny w wykazie MEN: ...**

# instakod

© Copyright by Instakod Sp. z o.o. 2019

ISBN 978-83-955025-0-7

Opracowanie merytoryczne i redakcyjne: **Ewa Gąsienica-Samek** (redaktor koordynator, redaktor merytoryczny), **Andrzej Gąsienica-Samek** (twórca języków programowania Assembly i Nianiolang), **zespół**

Konsultacja merytoryczno-dydaktyczna: **Joanna Śmigiełska, Katarzyna Olędzka**

Redakcja i korekta językowa: **Anna Stasiuk**

Projekt okładki, projekt graficzny, skład i łamanie: **Paweł Rupiński**

Autor zdjęć: **Ewa Gąsienica-Samek**

Autorzy rysunków: **Michał Ryczkowski i Ewa Gąsienica-Samek**

Instakod

02-587 Warszawa, ul. Wiktorska 17a

[www.instakod.pl](http://www.instakod.pl)

---

Nabyta przez Ciebie publikacja jest dziełem twórcy i wydawcy. Prosimy o przestrzeganie praw, jakie im przysługują. Zawartość publikacji możesz udostępnić nieodpłatnie osobom bliskim lub osobiście znanym, ale nie umieszczaj jej w internecie. Jeśli cytujesz jej fragmenty, to nie zmieniaj ich treści i koniecznie zaznacz, czyje to dzieło. Możesz skopiować część publikacji jedynie na własny użytek.

Szanujmy cudzą własność i prawo.

Dowiedz się więcej na [www.legalnakultura.pl](http://www.legalnakultura.pl)

## Spis treści

<b>WPROWADZENIE .....</b>	<b>5</b>
1 Bezpieczna praca z komputerem.....	7
2 Korzystanie z internetu.....	11
3. Na początku były dane – zera i jedyńki. ....	14
<b>PODSTAWY PROGRAMOWANIA.....</b>	<b>19</b>
1 Wejście i wyjście programu.....	21
2 Wyjście z programu, stałe i zmienne.....	36
3 Operacje arytmetyczne 1. ....	41
Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę.	
4 Operacje arytmetyczne 2. ....	48
Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę. Krok dalej.	
5 Operacje arytmetyczne 3.....	55
Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka.	
6 Operacje arytmetyczne 4.....	64
Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka. Krok dalej.	
7 Suma i różnica 1. ....	72
Suma i różnica w pudełku pomocniczym. Przypisanie wartości do pudełka.	
8 Suma i różnica 2.....	80
Suma i różnica w pudełku pomocniczym. Krok dalej.	
9 Instrukcja warunkowa 1.....	87
Pierwsze starcie. Jeżeli warunek jest spełniony, to podejmij działanie, w przeciwnym razie zakończ program.	

10	Instrukcja warunkowa 2.....	97
	Krok dalej. Jeżeli warunek jest spełniony, to podejmij działanie I, w przeciwnym razie podejmij działanie II.	

## **TECHNOLOGIE INFORMACYJNE..... 109**

1	Edytor graficzny.....	111
	Projekt międzyprzedmiotowy.	
	Rysowanie linii i krzywych.	
	Rysowanie kształtów. Powielanie wzorów. Pola tekstowe.	
2	Edytor tekstu.....	125
	Projekt międzyprzedmiotowy.	
	Dokumenty tekstowe. Polskie litery, znaki specjalne, skrótów klawiszowe.	
	Formatowanie czcionek. Listy numerowane.	
	Formatowanie tekstu, akapity, wcięcia, interlinie.	
3	Prezentacje.....	142
	Projekt grupowy międzyprzedmiotowy.	
	Dodawanie i usuwanie slajdu. Wstawianie tekstu.	
	Wklejanie ilustracji. Zasady tworzenia i tryb prezentacji.	
4	Arkusz kalkulacyjny.....	155
	Komórki, kolumny, wiersze i ich formatowanie.	
	Liczby w komórkach, operacje dodawania, odejmowania, dzielenia i mnożenia wartości dwóch komórek.	

# WPROWADZENIE

Witaj na lekcjach informatyki!

Czym będziesz się zajmować?

Poznasz zasady pracy przy komputerze (w szczególności w pracowni komputerowej) oraz dowiesz się o zaletach i zagrożeniach internetu.

Powszechny dostęp do sieci ułatwia każdemu z nas zdobywanie informacji, komunikację oraz własną ekspresję, ale niesie też ze sobą pewne niebezpieczeństwa.



Autor: Ewa Gąsienica-Samek

Większą część roku szkolnego poświęcisz na poznawanie podstaw programowania. Informatyka, podobnie jak matematyka, jest nauką ścisłą. Co więcej, na tych zajęciach, również będziesz rozwiązywać zadania. Istotną różnicą między lekcjami informatyki a matematyki jest to, że nie nauczyciel, a komputer sprawdzi poprawność twoich rozwiązań. Jeśli poprawnie rozwiążesz zadania, komputer poinformuje Cię o zaliczeniu. W przeciwnym razie odrzuci twoje rozwiązanie.

Pokażemy ci, jak korzystać z technologii informacyjnych do realizacji różnorodnych projektów z innych przedmiotów szkolnych, w tym do: pisania wypracowań w edytorze tekstu, wykonywania obliczeń w arkuszu kalkulacyjnym, tworzenia prezentacji oraz rysunków w edytorze graficznym.

Pomocnymi w czasie lekcji będą:



**Podręcznik**, w którym znajdziesz omówienie przedstawianych tematów, w tym przykłady i rozwiązania wzorcowe.

insta**Kod**

**Środowisko programistyczne**, w którym będziesz rozwiązywać zadania programistyczne. Znajdziesz je na stronie:

<http://assembly-lang.org>



**Zeszyt do informatyki** okaże się nieocenioną pomocą podczas rozwiązywania zadań programistycznych. Szkic wymyślnego rozwiązania powinien znaleźć się w zeszycie, zanim przystąpisz do kodowania na komputerze.

# 1

## Bezpieczna praca z komputerem

Na tej lekcji poznasz:

1. zasady prawidłowego zachowania w pracowni komputerowej;
2. zasady tworzenia i bezpiecznego korzystania z haseł;
3. zakres wiedzy i umiejętności, które zdobędziesz na lekcjach informatyki w 4 klasie.

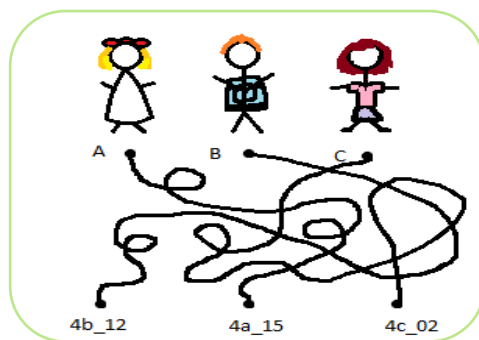
Naukę informatyki rozpoczniesz od zapoznania się z zasadami pracy z komputerem. Przeczytaj uważnie regulamin pracowni. Następnie przyjrzyj się swojemu komputerowi i zastanów się, czy umiesz nazwać wszystkie jego elementy.

### **Jak zachować się w pracowni?**

1. Stosuj się do poleceń nauczyciela. Pracuj tak, by nie przeszkadzać innym. Pamiętaj, jest to miejsce wyłożonej pracy. Pracuj w ciszy, a jeśli potrzebujesz pomocy, podnieś rękę i czekaj na nauczyciela. Pytania zadawaj ściszym głosem.
2. Dbaj o sprzęt komputerowy. Wszelkie usterki zgłaszaj nauczycielowi.
3. Na komputerze pracuj na swoim koncie. Na początku pracy zaloguj się - podaj login (nazwę użytkownika) oraz hasło. Kończąc pracę, wyloguj się.
4. Bez zgody nauczyciela nie wprowadzaj żadnych zmian na komputerze.
5. Zapisuj regularnie swoją pracę, by w razie awarii jej nie utracić.
6. Korzystaj z internetu jedynie do celów związanych z tematem zajęć.
7. Przestrzegaj wszelkich innych zasad określonych w regulaminie szkolnej pracowni komputerowej.

## Zadanie 1.

Zapisz w zeszyte do informatyki nazwy użytkowników uczniów A, B i C.



Rys.1

Hasła służą zabezpieczeniu kont i danych przed niepowołanym dostępem. Jeżeli z komputera korzysta więcej niż jedna osoba, każdy użytkownik powinien mieć swój login (niepowtarzalną nazwę) i hasło (ciąg znaków znanych tylko jemu).

Dobre, czyli bezpieczne hasło, składa się z wielkich i małych liter, cyfr oraz znaków specjalnych. Nie może być zbyt krótkie, ponieważ wtedy łatwo je odgadnąć. Przyjmuje się, że hasło powinno mieć co najmniej 8 znaków. Nie powinno zawierać liter tworzących łatwe do odgadnięcia wyrazy, w szczególności twojego imienia, nazwiska czy pseudonimu. Jeśli posiadasz więcej niż jedno konto, dobrze jest mieć osobne hasła do każdego z nich. Nikomu nie podawaj swojego hasła.

Jeżeli zauważysz, że z twoim kontem dzieje się coś podejrzanego, zgłoś to dorosłemu, któremu ufasz.

## Zadanie 2.

Przepisz poniższą tabelę do zeszytu do informatyki i oceń, czy podane hasła można uznać za bezpieczne. Wpisz przy każdym hasle TAK lub NIE oraz krótko uzasadnij odpowiedź. Pamiętaj, aby nie wpisywać rozwiązania w podręczniku.

ala123		Qwerty	
Pp0mM2jbgd!		Cgh56Bnj.	
Ktoragodzina		#234DfbGh	
12tomek		Ala2015	



## Zapamiętanie hasła

Hasło powinno być trudne do odgadnięcia przez innych, ale jednocześnie łatwe do zapamiętania przez siebie. Oto kilka wskazówek, które pomogą ci stworzyć i zapamiętać hasło:

Wskazówka 1. Pomyśl o jakimś powiedzeniu lub cytacie, który dobrze znasz. Gdy już wybierzesz cytat, weź pierwsze litery każdego ze słów i połącz je razem.

Przykład: Kiedy wrzesień, to już jesień, wtedy jabłek pełna kieszeń. → kwttjwjjpk

Wskazówka 2. Zmień niektóre litery na wielkie.

Przykład: dziwnynapis → dzlwnYnApls

Wskazówka 3. Zmień niektóre litery na cyfry lub znaki specjalne.

Przykład: karbolch → k@rb0lc!

### Zadanie 3.

Wybierz powiedzenie lub cytat dla każdego z przykładów. Przerysuj poniższą tabelę do zeszytu i uzupełnij ją. Zastosuj po kolei wszystkie trzy zasady i wygeneruj hasła. Nie podawaj swojego prawdziwego hasła. Pamiętaj, aby nie wpisywać rozwiązań w podręczniku.

Polecenie	Przykład 1	Przykład 2	Przykład 3
Napisz powiedzenie lub cytat.			
Wypisz jego pierwsze litery.			
Zmień niektóre litery na wielkie.			
Zmień niektóre litery na cyfry lub znaki specjalne. Możesz też dodać jakieś znaki.			

## Co będziemy robić na lekcjach informatyki?

Na pewno korzystałeś już z komputera do nauki i zabawy. Znasz podstawowe zasady użytkowania komputera. Na lekcjach informatyki rozwiniiesz swoje umiejętności techniczne o pisanie tekstów, przygotowywanie rysunków, a przede wszystkim o podstawy programowania. Wkroczysz w świat informatyki.

### Zadanie 4.

Dopasuj podane ikony , , ,  do oznaczonych zielonymi cyframi miejsc. Zapisz rozwiązanie w zeszycie do informatyki.

Na lekcjach **1** dowiesz się jak pisać **2**, przygotowywać **3**, a przede wszystkim nauczysz się programować. Wkraczasz w **4** informatyki.

## 2

# Korzystanie z Internetu

Na tej lekcji poznasz:

1. możliwości wykorzystania internetu;
2. zasady netykiety;
3. zasady bezpiecznego korzystania z internetu.

Internet to system połączeń między komputerami, który obejmuje niemal cały świat. Każde urządzenie w sieci ma swój internetowy adres (adres IP), który pozwala na jego identyfikację. Możliwości wykorzystania sieci są ogromne i ciągle rosną. Za jej pomocą możemy m.in.: komunikować się z rodziną i znajomymi, korzystać z materiałów dydaktycznych, robić zakupy, oglądać filmy i słuchać muzyki. Niestety nie wszystko to, co znajduje się w internecie, jest wartościowe. Naszym obowiązkiem jest nauczenie się, jak można korzystać z internetu w odpowiedzialny i bezpieczny sposób.

**Przeglądarka internetowa** to program komputerowy służący do pobierania i wyświetlania stron internetowych. Jeśli znasz adres strony, wystarczy wpisać go w pasku adresu i zatwierdzić klawiszem Enter. Przeglądarka wyświetli zawartość strony znajdującej się pod wskazanym adresem.

Gdy nie znasz adresu strony, możesz skorzystać z **wyszukiwarki**. Ten specjalny program przeszukuje strony internetowe i kataloguje je. Gdy użytkownik wpisze frazę, której szuka, wyszukiwarka, korzystając ze swoich zbiorów, szybko poda listę adresów stron pasujących do zapytania.

Poniżej znajdują się strony, z których będziemy korzystać:

- strona szkoły – znajdziesz na niej podstawowe informacje z życia szkoły (np. plan lekcji, opisy realizowanych projektów) oraz dokumenty, które określają jej funkcjonowanie (np. statut);
- strona miasta lub gminy – możesz dowiedzieć się o ciekawych wydarzeniach z twojej okolicy;
- strony edukacyjne dla uczniów z materiałami przydatnymi w nauce.

## Zadanie 1.

Podaj 5 adresów stron internetowych, z których warto korzystać.  
Krótko scharakteryzuj każdy z nich.

Przykład: <http://sieciaki.pl> – strona dla dzieci o bezpieczeństwie w sieci.

## Zasady netykiety

W życiu codziennym obowiązują zasady kulturalnego zachowania. Ich odpowiednikiem w sieci jest netykieta, czyli zbiór zasad kulturalnego i prawidłowego korzystania z internetu. Powinni jej przestrzegać wszyscy internauci.

## Zadanie 2.

Przykład netykiety forum internetowego.

Na pewnym forum internetowym zamieszczona została przykładowa netykieta. Niestety, w niektórych wyrazach pozamieniano kolejność liter. Jakie to wyrazy?

1. Bądź taktowny i kulturalny. **Sunzaj** ludzi, z którymi się komunikujesz.
2. Szanuj pracę innych ludzi. Możesz **ktazyorsć** z materiałów opublikowanych przez innych, ale zawsze podawaj ich **artuoa** i źródło.
3. Na forum pisz na temat. Staraj się, aby Twoje wypowiedzi były zwarte.
4. Zwracaj uwagę na ortografię. Poprawnie napisana wiadomość jest **biarezdj** wiarygodna.
5. Nie nadużywaj **wclkieih** liter i kolorów. Elementy graficzne nie powinny zajmować zbyt **wliee** miejsca.

## **Zasady korzystania z internetu**

1. Nie podawaj bez potrzeby w internecie swojego imienia i nazwiska. Lepiej posługiwać się nickiem, czyli internetowym pseudonimem.
2. Komunikując się przez internet z nieznaną osobą, nie wiesz, jakie ma intencje. Nie podawaj jej swojego telefonu, adresu domowego ani adresu szkoły, do której chodzisz.
3. W internecie znajdziesz treści, które mogą zarówno okazać się pożyteczne jak i takie, które są szkodliwe. Bądź odpowiedzialny w korzystaniu z internetu.
4. Kiedy ktoś lub coś cię zaniepokoi, koniecznie powiedz o tym rodzicom lub innej zaufanej osobie. Nie lekceważ sprawy.

### **Zadanie 3.**

Internetowe zasady.

Powyżej przedstawiono 4 zasady korzystania z Internetu. Dla każdej z nich opracuj znak graficzny.

### 3

## Na początku były dane – zera i jedynki

Zastanów się, do czego służy ci komputer, tablet czy smartfon.

Zapewne udzieliłeś/udzieliłaś odpowiedzi podobnej do wymienionych poniżej:

- do oglądania filmów i słuchania muzyki,
- do grania,
- do kontaktowania się z ludźmi, m.in. na forach społecznościowych,
- do wyszukiwania informacji potrzebnych do odrobienia pracy domowej.

Na lekcjach informatyki dowiesz się, w jaki sposób te urządzenia działają. Urządzenia elektroniczne przechowują ogromne ilości danych. Miejsce, w którym są one przechowywane, zwane jest pamięcią.

Zastanów się nad poniższymi pytaniami.

1. W jaki sposób wszystkie dane zapisane są w pamięci komputera?
2. W jaki sposób komputer przetwarza dane?

Na pierwsze pytanie odpowiesz już dziś. Na drugie zaś, odpowiedź będziesz poznawać tydzień po tygodniu, miesiąc po miesiącu, rok po roku, ucząc się pisania programów przetwarzających dane liczbowe i tekstowe.

Komputer jest urządzeniem elektronicznym. Pamięć komputera składa się z komórek pamięci. Prąd płynący przez znajdujący się w środku komputera przewód reprezentujemy jako **1**. Brak prądu reprezentujemy jako **0**. Stany komórki pamięci komputera możemy przedstawić następująco:

0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

### Ciekawostka

System pozycyjny używany do opisu stanu poszczególnych komórek pamięci komputera to **system binarny**. Wszystkie dane zapisywane są jedynie za pomocą cyfr 0 i 1. Przedrostek „bi” oznacza dwa - w tym przypadku korzystający z dwóch cyfr - zera i jedynki. Inną nazwą systemu binarnego jest system dwójkowy.

Podstawową jednostką informacji jest **bit**. Ilość bitów mówi nam, ile danych może zmieścić się na dysku.





### Ciekawostka

Cyfry 0 i 1 nazywane są cyframi binarnymi, po angielsku *binary digits*. Nazwa **bit** wzięła się ze skrótu słów:

**binary + digit = bit**

8 bitów nazywamy **bajtem**.

Poniżej zestawiono przykładową pojemność kilku typów plików oraz dysków:

	plik tekstowy z wypracowaniem na zajęcia z języka polskiego	15 KB, czyli 15 kilobajtów	to jest około 15 000 bajtów
	zdjęcie zrobione twoim telefonem	3 MB, czyli 3 megabajtów	to jest około 3 000 000 bajtów
	pojemność uczniowskiego pendrive'a	16 GB, czyli 16 gigabajtów	to jest około 16 000 000 000 bajtów
	pojemność dysku półprzewodnikowego	0.5 TB, czyli pół terabajta	to jest około 500 000 000 000 bajtów

Można zatem stwierdzić, że komputer zapisuje informacje w biliardach komórek pamięci tylko za pomocą dwóch stanów, które opisujemy zerami i jedynekami.

Postawmy prostsze pytanie: W jaki sposób komputer zapisuje liczby?

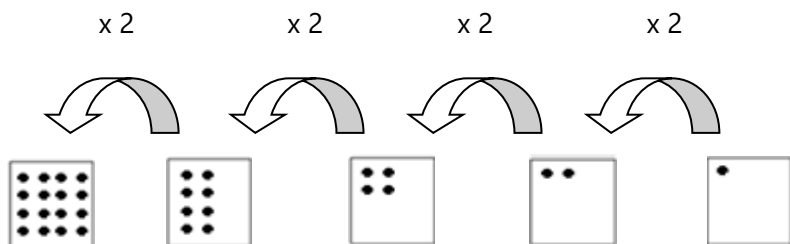
Weźmy liczbę 12. Jej zapis w systemie dziesiętnym, z którego korzystamy na co dzień, to jedna dziesiątka i dwie jedności.

$$12 = 1 \times 10 + 2 \times 1$$

Zapis ten możemy przełożyć na gotówkę. Dwanaście złotych mogą stanowić na przykład banknot dziesięciozłotowy i dwie monety jednozłotowe.

Ale jak zapisać 12 zł wyłącznie za pomocą zer i jedynek?

Pomogą nam w tym innego typu banknoty, które nazwiemy kartami systemu binarnego. Pierwsza karta od prawej zawiera jedną kropkę odpowiadającą 1 zł, a każda kolejna dwa razy więcej kropek niż poprzednia. Mamy więc karty: 1 zł, 2 zł, 4 zł, 8 zł, 16 zł, itd.



Które z nich należy wybrać, by łącznie było na nich 12 kropek?


Wybieramy kartę z 8 kropkami oraz kartę z 4 kropkami, ponieważ

$$8 + 4 = 12$$

Karty ustawione są od najmniejszej na prawo do największej na lewo. W miejscach kart, które zostały wybrane, wpisujemy „1”, a w pozostałych miejscach „0”.

W ten sposób powstał zapis liczby 12 w systemie binarnym. Jest to 01100.



Liczba w systemie dziesiętnym	Reprezentacja na kartach systemu binarnego	Zapis w systemie binarnym					
12		<table border="1" data-bbox="765 287 1102 360"> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </table>	0	1	1	0	0
0	1	1	0	0			

Zapiszmy większą liczbę, na przykład 28.

Mamy do dyspozycji karty systemu dwójkowego. Wiadomo, że każda kolejna powstaje z poprzedniej mnożonej przez 2. Ustawione są od najmniejszej z prawej strony do największej z lewej strony. Możemy tworzyć kolejne większe karty, jeśli jest taka potrzeba. Np. kolejną większą kartą od 512 będzie 1024.

512	256	128	64	32	16	8	4	2	1
-----	-----	-----	----	----	----	---	---	---	---

Bierzemy największą kartę, jaka mieści się w 28. Jest to karta 16.

Z 28 kropek pozostaje nam jeszcze 12 kropek (ponieważ  $28 - 16 = 12$ ).

Znów bierzemy największą kartę, jaka mieści się w 12. Jest to karta 8.

Pozostają jeszcze 4 kropki (ponieważ  $12 - 8 = 4$ ), więc odpowiednią kartą będzie 4.

28 reprezentujemy kartą 16, kartą 8 oraz kartą 4.

512	256	128	64	32	16	8	4	2	1
-----	-----	-----	----	----	----	---	---	---	---

Jej zapis w systemie binarnym wygląda więc tak.

0	0	0	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

### Zadanie 1.

Zapisz liczby: 2, 4, 8, 16, 32 w systemie binarnym.

### Zadanie 2.

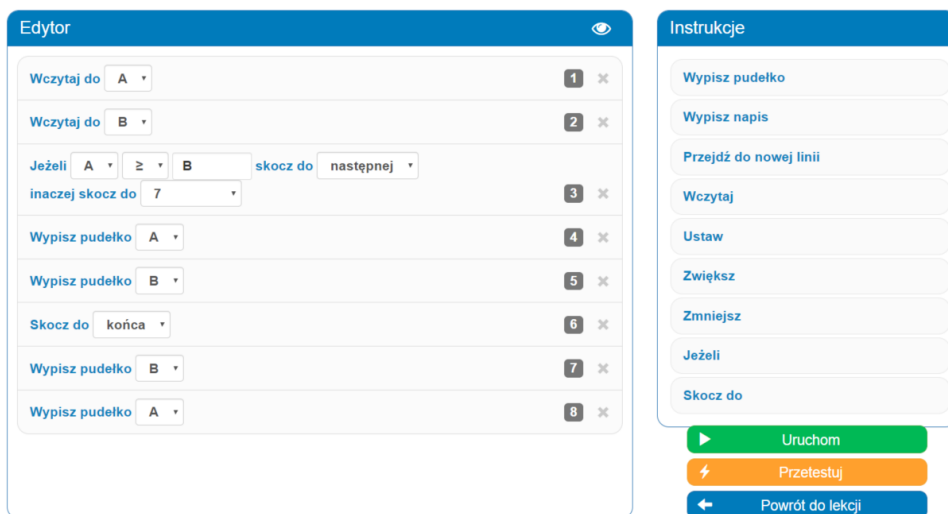
Zapisz liczby: 1, 3, 7, 15, 31 w systemie binarnym.

### Zadanie dla mistrza

Jaką zależność można zauważyć między zapisem liczb z zadania 1 i zadania 2? Jak będzie wyglądał zapis liczby 63?



# PODSTAWY PROGRAMOWANIA



Przed tobą 10 rozdziałów, w których zapoznasz się z podstawami programowania.

Dowiesz się: czym jest zmienna, co to jest wejście i wyjście programu, czym jest stan wartości zmiennych, jak wykonywać operacje arytmetyczne na zmiennych oraz co to jest instrukcja warunkowa.

W każdym rozdziale znajdziesz omówienie wprowadzanych pojęć, wzorcowe rozwiązania przykładów oraz wykaz zadań programistycznych do rozwiązania w środowisku programistycznym InstaKod, które znajdziesz na stronie:

<http://assembly-lang.org>

Do dzieła!



# InstaKod 1

## Wejście i wyjście programu

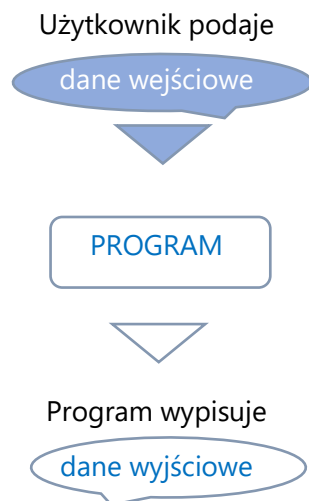
W poprzednim rozdziale dowiedziałeś się/dowiedziałas się, jak zapisywane są dane liczbowe w pamięci komputera. Czas bliżej przyjrzeć się temu, w jaki sposób komputer przetwarza dane.

Oczywiście nic nie dzieje się samo. To programista tworzy program, który przetwarza dane wejściowe A, by otrzymać dane wyjściowe B. Programistę można porównać do kierowcy autobusu, który ma zawieźć pasażerów z punktu A do punktu B.

Aby bezpiecznie i szybko zawieźć pasażerów na miejsce, kierowca musi znać trasę. W przeciwnym razie, zamiast w górę, zawiezie swoich pasażerów nad morze albo wjedzie na drogę gruntową i zakopie się w błocie. Musi też wiedzieć, jak prowadzić autobus, czyli na jakie działania zareaguje kierowany przez niego pojazd. Jeśli, przykładowo, kierowca podskoczy na siedzeniu, autobus nie pojedzie do przodu. Wie natomiast, że czasem musi słabiej lub mocniej nacisnąć pedał gazu, skręcić kierownicą w prawo, odbić w lewo lub zahamować.

Podobnie jak z kierowcą autobusu jest z programistą. Musi on najpierw wymyślić trasę, czyli algorytm rozwiązania danego problemu. Następnie musi przetłumaczyć swoje rozwiązanie na język zrozumiały dla komputera, czyli kod.

Algorytm to zapis wymyślonego przez nas rozwiązania. Jest listą kroków do wykonania zapisanych w zrozumiały dla nas sposób. Jest przepisem, w jaki sposób z podanych danych wejściowych otrzymać pożądane dane wyjściowe.

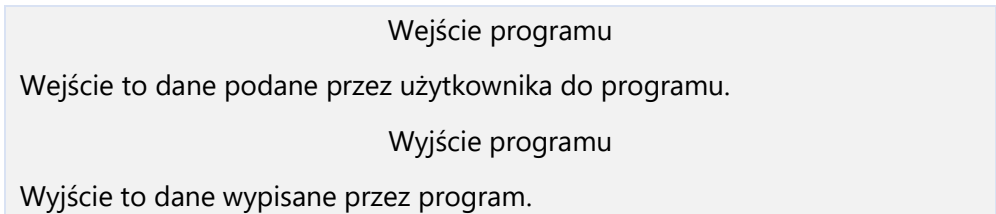


Implementacja to napisanie programu, czyli zapisanie rozwiązania w języku programowania.

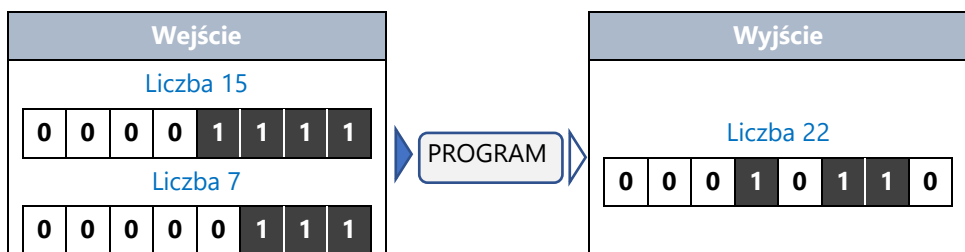
Kim są pasażerowie w przypadku komputera?

Pasażerem jest użytkownik, który uruchamia napisany przez programistę program. Pasażer, który wsiada w punkcie A, oczekuje, że zostanie dowieziony do punktu B. Tak samo zachowuje się użytkownik. Uruchamiając program, na wejściu podaje dane wejściowe A i oczekuje, że po przetworzeniu przez program na wyjściu otrzyma wynik B.

Będziemy więc uczyć się pisać programy, które będą wczytywać liczby i wypisywać wartości końcowe.



Przykładem może być program, który wczytuje dwie liczby i wypisuje ich sumę. Jeśli na wejściu podamy 15 i 7, to na wyjściu otrzymamy sumę 22. Komputer pobierze liczby zapisane w systemie dziesiętnym i zapisze obie liczby w pamięci w systemie binarnym. Następnie w zapisie binarnym wykona operację dodawania i wynik zapisze w pamięci również w systemie binarnym. Dopiero na wyjściu zapis binarny wyniku zostanie przeliczony na zapis w systemie dziesiętnym i wyświetlony na konsoli. O tym, jak wykonuje się niektóre operacje arytmetyczne, w szczególności dodawanie w systemie binarnym, będziemy uczyć się w klasie 6.



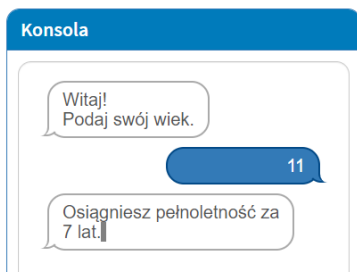
## Wyjście stałe, czyli komunikaty z programu. Instrukcja „Wypisz napis”

Poznaj robota Bajtka. Bajtek niestety nie rozumie mowy ludzi, natomiast dobrze zna język Assembly (czyt. asembli). Żeby się porozumieć z Bajtkiem, musisz rozmawiać z nim kodem w języku Assembly. Jeśli zaprogramujesz go odpowiednio, to Bajtek wykona ustalone przez siebie zadania. W ustawionych na półce pudełkach A, B, C i D Bajtek przechowuje liczby i może wykonywać na nich proste operacje. Bajtek może wchodzić w interakcje z twoimi kolegami, wyświetlać im komunikaty i wkładać do pudełek podane przez nich liczby. Na wyświetlaczu (który będziemy nazywać konsolą) Bajtek wyświetla zapis interakcji z twoimi kolegami. Od tej pory kolegów wchodzących w interakcje z Bajtkiem będziemy nazywać użytkownikami.

KOMPUTER



BAJTEK



UŻYTKOWNIK

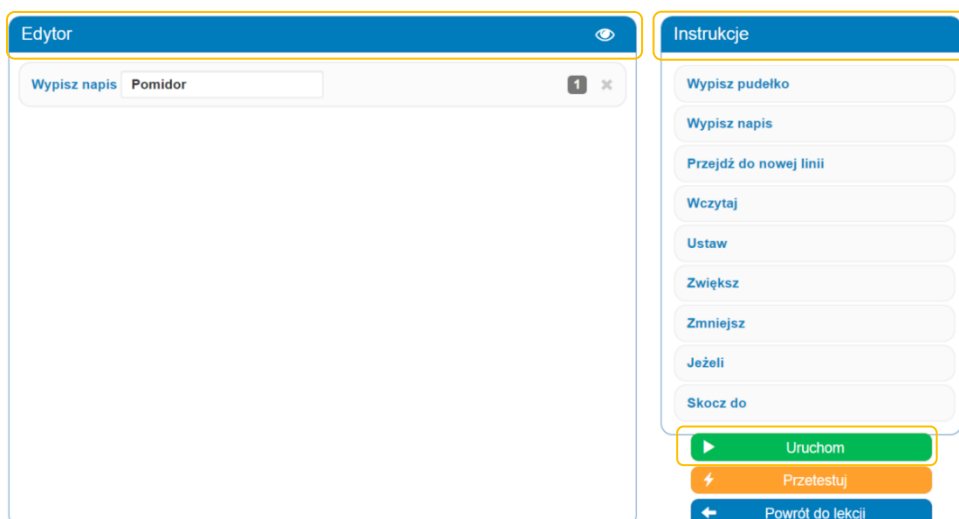


ZOSIA

Naukę programowania rozpoczniemy od tworzenia programów wczytujących liczby i wypisujących liczby oraz słowa.

Środowisko, z którego będziemy korzystać, Assembly, dostępne jest na stronie [assembly-lang.org](http://assembly-lang.org). W polu „Edytor” będziesz tworzyć swoje programy, przeciągając instrukcje z mniejszej ramki.

Napišemy teraz nasz pierwszy program. Przeciągnij do edytora instrukcję **Wypisz napis**. Instrukcja ta służy do wypisywania na ekran tekstu z białego pola. Tekst wpisujemy z klawiatury, która pojawia się na ekranie lub bezpośrednio z klawiatury komputera. Wpisz słowo *Pomidor*.

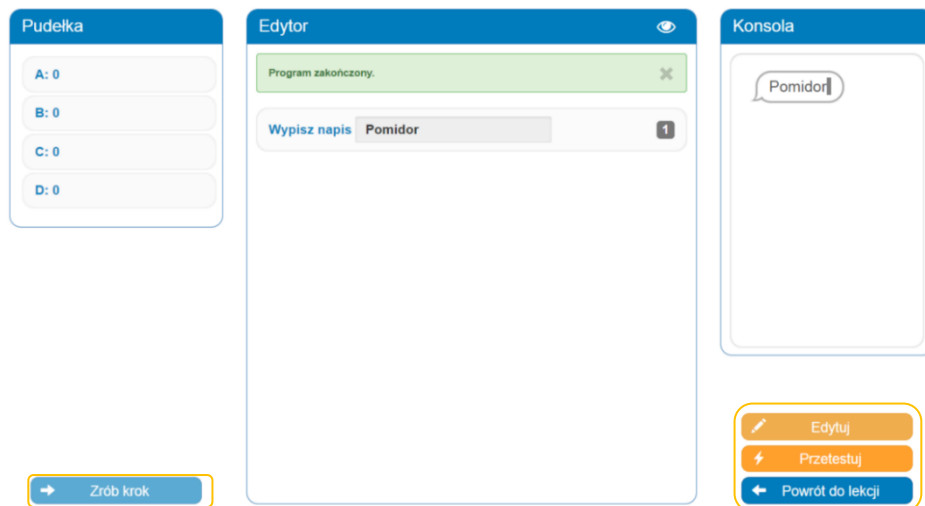


Gratulacje! Powstał właśnie twój pierwszy program.

Możesz teraz przetestować jego działanie. Aby przejść do trybu uruchomienia programu, kliknij zielony przycisk [Uruchom](#).

Po prawej stronie twojego kodu pojawi się ramka [Konsola](#). W tym miejscu komputer będzie się z tobą komunikował. Dostępne w tym widoku przyciski to: [Zrób krok](#), [Powrót do lekcji](#), [Edytuj](#), [Przetestuj](#).

[Przycisk Zrób krok](#) służy do wykonania pojedynczej instrukcji.



Po kliknięciu [Zrób krok](#) na konsoli pojawia się napis *Pomidor*. W okienku edytora w zielonej ramce po wykonaniu ostatniej instrukcji programu pojawi się komunikat [Program zakończony](#).



Gdy wybierzesz przycisk **Edytuj**, wrócisz do trybu pisania programu.

W trybie edytowania programu jesteś kierowcą, czyli programistą, który tworzy program. Po naciśnięciu **Uruchom**, stajesz się na chwilę pasażerem, czyli użytkownikiem programu. Testujesz go, sprawdzając, czy działa jak należy.

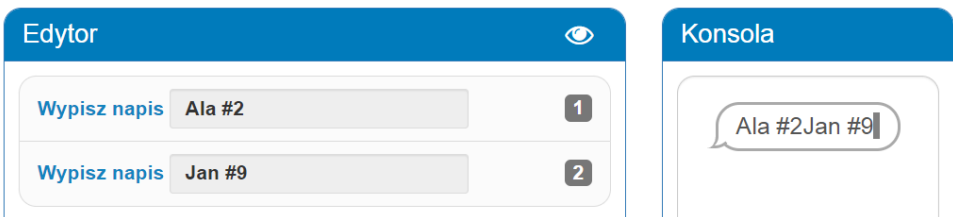
## Instrukcja „Przejdź do nowej linii”

Załóżmy, że chcemy, by na konsoli w oddzielnych liniach pojawiły się dwa napisy (składające się z liter, cyfr i symboli):

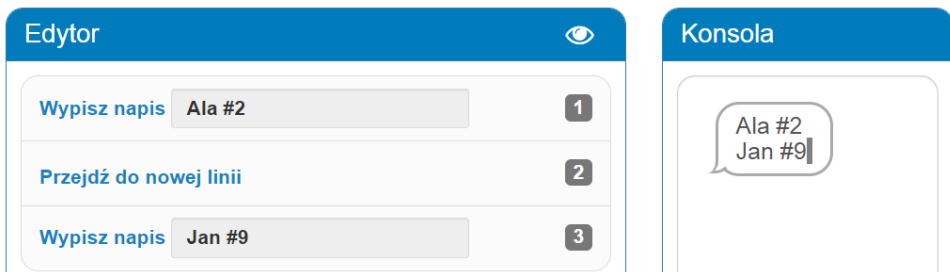
Ala #2

Jan #9

Jeśli użyjemy dwukrotnie instrukcji „Wypisz napis”, to na konsoli pojawi się w jednym wierszu napis *Ala #2Jan #9*.



Nie o to nam chodziło. Chcemy, by po wypisaniu napisu *Ala #2*, program przeszedł do nowej linii i wypisał *Jan #9*. W tym celu używamy instrukcji „Przejdź do nowej linii”.



Jeśli program zawsze wypisuje to samo, niezależnie od tego, jakie dostanie dane wejściowe (pytanie), mówimy, że jest to **wyjście stałe**. Przypomina ono trochę grę w pomidora. W tej grze, na każde pytanie zawsze odpowiadamy *Pomidor*. Komputer nigdy nie przegrałby w takiej grze. Wystarczyłoby zbudować program, używając instrukcji **Wypisz napis** `Pomidor`.

## Pudełko. Wejście. Instrukcja „Wczytaj”

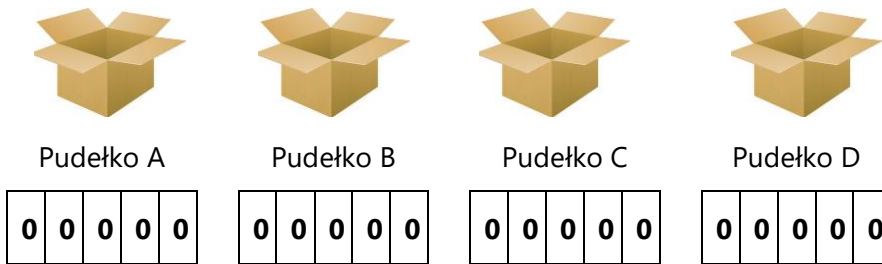
Wiesz już, jak zaprogramować Bajtka by wypisywał różne komunikaty takie jak na przykład: *Witaj Janku, 786, #Kasia* czy *Podaj liczbę*.



Witaj Janku!  
578  
#Kasia  
Podaj liczbę. ||

Dowiesz się teraz, jak programować Bajtka by przetwarzał dane, czyli by przyjmował dane wejściowe, wykonał na nich zaplanowane przez siebie operacje i zwrócił dane wyjściowe. Dane wejściowe komputer Bajtek zapisuje w pudełkach. Pudełko to miejsce w pamięci komputera, w którym przechowywana jest liczba.

W Assembly Bajtek ma do dyspozycji 4 pudełka. W każdym z nich może zapisać jedną liczbę. Rozmiar każdego pudełka jest ściśle określony przez liczbę komórek pamięci do niego przypisanych.

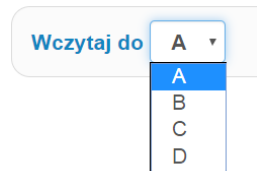


W Assembly do każdego pudełka przypisanych jest 16 komórek pamięci. W każdej komórce pamięci możemy zapisać jeden z dwóch stanów reprezentowanych przez cyfry 0 i 1. Wiesz już, że przy pomocy cyfr 0 i 1 możesz zapisać dowolną liczbę. Zastanów się, jak komputer zapisze liczbę 7. Czy wiesz, jaką największą liczbę możesz zapisać w 16 komórkach pamięci?

Pudełka	
A:	0
B:	0
C:	0
D:	0

Na samym początku, gdy program zostanie uruchomiony, w każdym z pudełek Bajtek ma zapisaną liczbę 0. O tym, jaka liczba w kolejnych krokach wykonywania programu znajduje się w danym pudełku, informuje nas ramka [Pudełka](#).

Instrukcja „Wczytaj” umożliwi użytkownikowi podanie liczby do programu, a Bajtek zapisze ją w wybranym w instrukcji pudełku. Do wyboru mamy pudełka: A, B, C i D.



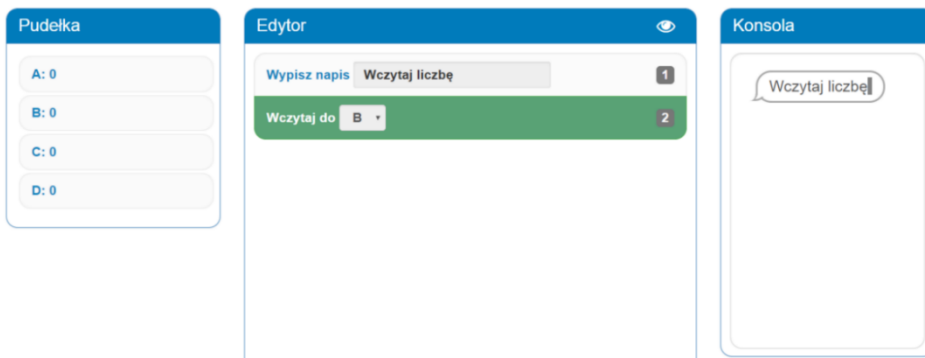
### Zmienna, inaczej pudełko

Pudełko to miejsce w pamięci komputera, w którym przechowywana jest liczba. Każde pudełko ma swoją nazwę oraz rozmiar (maksymalną wielkość związaną z liczbą komórek pamięci przypisaną do pudełka). W Assembly dostępne są pudełka: A, B, C i D, z których każde ma rozmiar 16 komórek pamięci (bitów) oraz przypisaną wartość początkową 0.

Założmy, że chcemy by Bajtek poprosił użytkownika o podanie liczby i zapisał ją w pudełku B.

Bajtek wyświetli napis *Wczytaj liczbę* jeśli podamy mu instrukcję `Wypisz napis Wczytaj liczbę`. Za pomocą instrukcji `Wczytaj do B` umożliwiamy użytkownikowi podanie liczby, którą Bajtek zapisze w pudełku B.

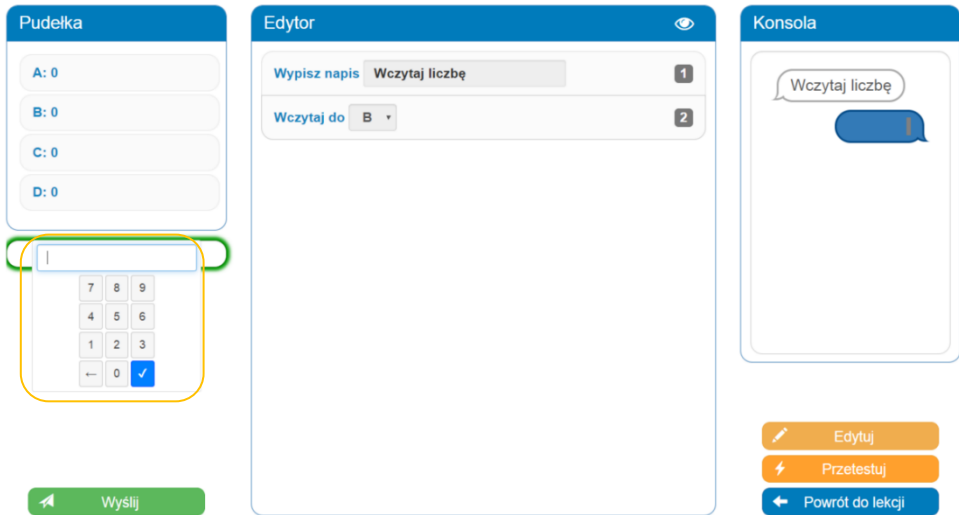
Uruchommy poniższy program i wykonajmy pierwszy krok. Na konsoli pojawia się szary dymek „rozmowy” komputera Bajtka, na którym jest napis *Wczytaj liczbę*.



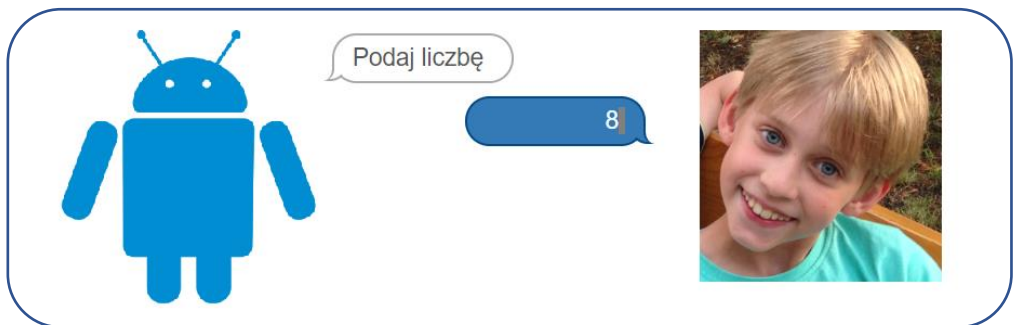
Zróbmy teraz kolejny krok. Na konsoli pojawia się niebieski dymek rozmowy użytkownika. W ten sposób Bajtek umożliwia użytkownikowi podanie liczby.

Równoległe, pod ramką **Pudełka**, pojawia się klawiatura, na której możemy wpisywać liczby.

Wybermy liczbę 8.

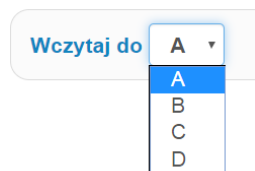


Po wczytaniu liczby zaktualizowane zostaną wartości pudełek. W ramce „Pudełka”, obok pudełka B, znajdzie się teraz liczba 8. Informacja o podanej przez użytkownika liczbie 8 pojawi się równoległe w niebieskim dymku na konsoli.



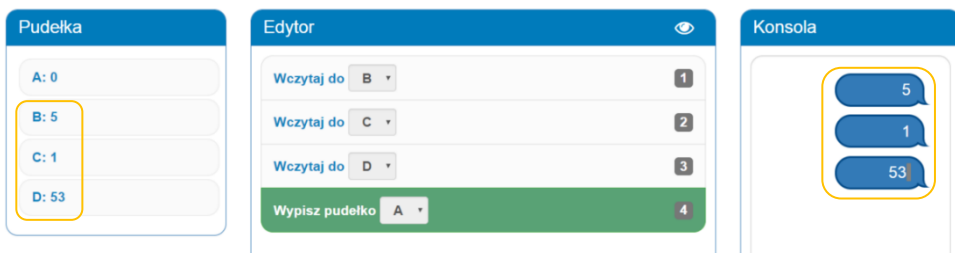
## Wyjście - Instrukcja „Wypisz pudełko”.

Gdy chcemy by Bajtek pokazał na konsoli, jaka liczba aktualnie znajduje się w danym pudełku, używamy instrukcji „Wypisz pudełko”. Zwróć uwagę, że możemy wybrać jedno z 4 pudełek: A, B, C lub D.

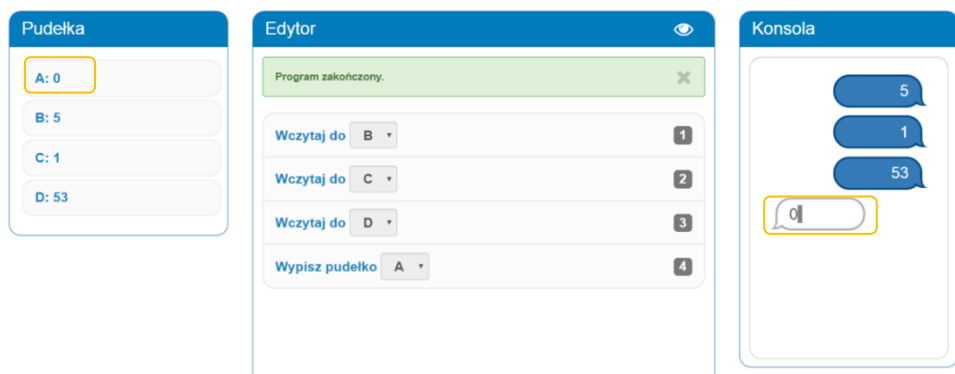


Załóżmy, że chcemy napisać program, który wczyta liczby do pudełek: B, C i D oraz wypisze wartość pudełka A.

Uruchommy poniższy program i wczytajmy następujące liczby: 5, 1 i 53. Po każdym kroku wartości pudełek w ramce „Pudełka” są aktualizowane, a na konsoli pojawia się informacja, jakie liczby zostały wczytane.



Pozostało jeszcze wykonanie ostatniego kroku. Uruchamiamy instrukcję **Wypisz pudełko A**. Na konsoli pojawi się liczba aktualnie znajdująca się w pudełku A. W naszym przykładzie wypisana zostanie liczba 0.



Jaka jest różnica między instrukcjami „Wypisz napis”, a „Wypisz pudełko”?

**Wypisz napis** Pomidor – za każdym razem przy uruchomieniu program wypisze na konsoli komunikat *Pomidor*. **Wypisz pudełko** B – wypisze liczbę, która w danym momencie wykonywania programu znajduje się w pudełku B.

## Wyjście z programu, stałe i zmienne

Wyjście to teksty i dane wypisane przez Bajtka na konsoli w celu komunikacji z użytkownikiem.

W Assembly mamy trzy instrukcje wyjścia. Są to:

**Wypisz pudełko** C wypisuje na konsoli liczbę znajdującą się w pudełku C. Wypisana wartość jest zmienna i zależy od tego, jaka liczba aktualnie znajduje się w pudełku C.

**Wypisz napis** Ala 45# wypisuje na konsoli tekst z ramki. Służy programiście do wyświetlania napisów dla użytkownika programu. Wypisywany napis (liczba, znak, słowo czy symbol) jest stały i nie zależy od wczytanych przez użytkownika danych.

**Przejdź do nowej linii** sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.

## Jak tworzone są programy?

Jak powstają duże programy, takie jak gry komputerowe czy aplikacje?

Aplikacje są odpowiedzią na potrzeby klientów (lub też innych pomysłodawców). Każdy klient posiada wizję tego, czego oczekuje od aplikacji. Zrozumienie oczekiwań klienta jest jednym z pierwszych kroków, jakie podejmuje zespół programistów odpowiedzialny za jej stworzenie.

Zwykle zaczyna się od opracowania założeń. Tworzy się opis, jak aplikacja ma działać, a następnie „rozwiązanie”, czyli plan tego, jak aplikacja ma być skonstruowana. Następnie przechodzi się do implementacji (pisania kodu programu). Już w trakcie pisania kodu testuje się program, by na bieżąco poprawiać błędy. Weryfikowane są także pierwotne założenia i rozwiązania. Implementowane są zmiany. Proces jest powtarzany dopóki, dopóty nie zostanie uzyskany zadowalający efekt.

Proste programy, jakie będziesz tworzyć, by rozwiązać przypisane Ci zadania informatyczne, powstawać będą według podobnego schematu 😊.

## Schemat rozwiązania zadania

A jak ty będziesz tworzyć swoje programy? Oto kilka przydatnych wskazówek:

- przeczytaj ze zrozumieniem treści zadania,
- wymyśl liczby, które podasz do programu (wejście) i policz oczekiwany wynik, czyli wyjście,
- wymyśl rozwiązanie,
- zapisz rozwiązanie w formie diagramu w zeszycie,
- stwórz kod (tworzenie kodu nazywamy implementacją),
- uruchom program dla założonego wejścia i porównaj wyjście z oczekiwanym,
- przetestuj,
- przeanalizuj wyniki testów i błędy,
- popraw,
- testuj i poprawiaj aż do skutku.

### Przykład 1.

Napisz program, który wypisze na konsoli literę *a*. Następnie w nowej linii wypisze napis *bbb*.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.




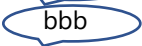
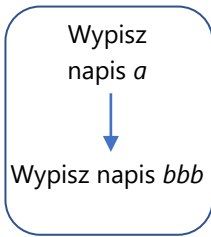
Wejście i wyjście	Konsola	
W niebieskich dymkach po prawej stronie pojawiają się podane przez użytkownika liczby, zaś w białych dymkach po lewej stronie pojawia się odpowiedź komputera.	Wyjście	Wejście
		
Użytkownik nie podaje żadnych danych do programu. Komputer za każdym razem wypisuje ten sam komunikat, literę <i>a</i> i w oddzielnym wierszu napis <i>bbb</i> .	 	

Diagram rozwiązania i jego implementacja.

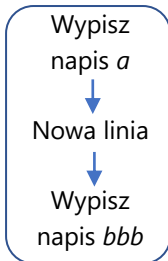


Zwróć uwagę, że przy każdym uruchomieniu programu na konsoli ma się pojawić ten sam tekst. Mówimy, że jest to wyjście stałe. Do wypisywania tekstu służy instrukcja „Wypisz napis”.

```
Wypisz napis a
Wypisz napis bbb
```

Po uruchomieniu programu na konsoli pojawia się napis: `abbb`.

Nie jest to jeszcze poprawne wyjście. Zgodnie z poleceniem napisy powinny być wypisane w osobnych liniach.



Pomiędzy instrukcjami „Wypisz napis” umieszczamy instrukcję „Przejdź do nowej linii”.

```
Wypisz napis a
Przejdź do nowej linii
Wypisz napis bbb
```

## Przykład 2.

Napisz program, który wczyta liczbę do pudełka B i wypisze jego wartość, a następnie ponownie wczyta liczbę do pudełka B i wypisze jego wartość.

Dzielimy zadanie na kroki, które kolejno musimy wykonać.

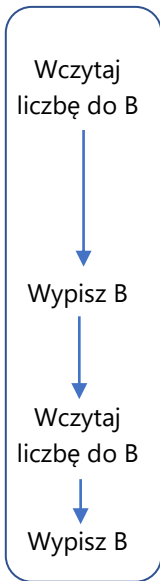
Wczytaj liczbę do pudełka B | wypisz jego wartość (z pudełka B) |  
ponownie wczytaj liczbę do pudełka B | wypisz jego wartość.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 1 i 5 program powinien na wyjściu wypisać 1 i 5.	

Diagram rozwiązania i jego implementacja.





Pierwszym krokiem jest wczytanie liczby. Do wczytania używamy zawsze instrukcji „Wczytaj”. Mamy możliwość wybrania jednego z czterech pudełek: A, B, C lub D. Zgodnie z poleceniem wybieramy pudełko B.

Wczytaj do B

Podana liczba trafia do pudełka B. Aby liczba z pudełka B została wypisana na konsoli, musimy użyć instrukcji `Wypisz pudełko B`.

Wypisz pudełko B

Ponownie wczytujemy liczbę do pudełka B i wypisujemy ją na konsolę. Zastanów się, co się stało z poprzednią wczytaną liczbą.

Wczytaj do B

Wypisz pudełko B

### Przykład 3.

Napisz program, który wczyta liczbę do pudełka C i wypisze pudełko C. Następnie przejdzie do nowej linii, ponownie wypisze wartość pudełka C oraz literę *a*.

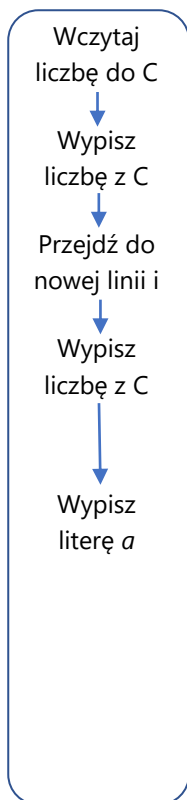
Dzielimy zadanie na kroki, które kolejno musimy wykonać.

Wczytaj liczbę do pudełka C | wypisz pudełko C | przejdź do nowej linii | wypisz wartość pudełka C | wypisz literę *a*.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 4 program powinien na wyjściu wypisać w oddzielnych liniach 4 oraz 4a.	<p>The terminal output shows the number '4' on the first line and '4a' on the second line. The input '4' is shown in a blue oval, and the outputs '4' and '4a' are also circled in blue.</p>

## Diagram rozwiązania i jego implementacja.



Na początku musimy wczytać i wypisać liczbę. Zgodnie z poleceniem użyjemy pudełka C. Wczytujemy i wypisujemy liczbę, pamiętając, że zapisana jest w pudełku C.

```
Wczytaj do C
```

```
Wypisz pudełko C
```

Zgodnie z poleceniem przechodzimy do nowej linii i wypisujemy jeszcze raz wartość pudełka C.

```
Przejdź do nowej linii
```

```
Wypisz pudełko C
```

Wypisujemy literę *a*. Której z dwóch instrukcji wyjścia służących do wypisywania na konsoli użyjemy?

```
Wypisz napis a
```

„Wypisz napis” – zawsze wypisuje napis podany w instrukcji.

„Wypisz pudełko” – wypisuje wartość, która w danej chwili jest w pudełku.

Używamy instrukcji „Wypisz napis”.

## Zadania InstaKod. Rozdział 1. Wejście i wyjście programu

### Poziom 1

- IV 1.1 Napisz program, który wczyta liczbę do pudełka A.
- IV 1.2 Napisz program, który wypisze literę  $x$ .
- IV 1.3 Napisz program, który wypisze napis *Rozdział*.

### Poziom 2

- IV 1.4 Napisz program, który wczyta liczbę do pudełka B i ją wypisze.
- IV 1.5 Napisz program, który wczyta liczbę do pudełka A i wypisze znak  $a$ . Następnie w nowej linii wypisze wczytaną liczbę.
- IV 1.6 Napisz program, który wypisze literę  $b$ , wczyta liczbę do pudełka C i wyświetli ją na ekranie.

### Poziom 3

- IV 1.7 Napisz program, który wczyta liczbę do pudełka A. Następnie wypisze literę  $c$ , wczytaną liczbę oraz w nowym wierszu trzy literki  $a$ .
- IV 1.8 Napisz program, który wczyta liczbę do pudełka B i wypisze jego wartość. Następnie ponownie wczyta liczbę do pudełka B i wyświetli jego nową wartość.
- IV 1.9 Napisz program, który wczyta liczbę do pudełka D oraz wypisze napis *Liczby*. Następnie w osobnych wierszach wypisze trzykrotnie wczytaną liczbę.

### Zadania dla mistrza

- IV 1.10\* Napisz program, który wypisze na konsoli trójkąt prostokątny złożony z gwiazdek o wysokości 4 i podstawie 7.
- IV 1.11\* Mrówki porozumiewają się specjalnym kodem. Początek istotnej wiadomości zaczynają od literki  $x$ , a w nowym wierszu, dla większego bezpieczeństwa, dwukrotnie zapisują ważne liczby. Napisz program, który wczyta do pudełka D, ile nasion udało się zgromadzić mrówkom, a następnie wypisze tę informację zgodnie z sekretnym kodem ważnych wiadomości.

## InstaKod 2

### Wyjście programu, stałe i zmienne

Przypomnijmy podstawowe instrukcje, które rozumie Bajtek.

#### Wejście programu

Wejście to liczby, jakie użytkownik podaje komputerowi Bajtkowi, wpisując je na klawiaturze.

**Wczytaj do B** prośba do Bajtka by użytkownik mógł podać mu liczbę, którą Bajtek zapisze w pudełku B, kasując liczbę, która była tam wcześniej zapisana.

#### Wyjście programu, stałe i zmienne

Wyjście to wszystko to, co Bajtek przekazuje użytkownikowi.

Wyjście stałe jest zawsze takie samo, niezależne od wejścia.

Wyjście zmienne jest różne, zależne od podanych danych wejściowych.

**Wypisz pudełko C** Bajtek zajrzy do pudełka C i wypisze na konsoli liczbę znajdującą się aktualnie w pudełku C; jest to wyjście zmienne.

**Wypisz napis Ala 45#** Bajtek wypisze na konsoli tekst z ramki. W ten sposób możesz za pomocą Bajtka przekazywać komunikaty do twojego kolegi. Pamiętaj, że wypisywany komunikat jest stały i nie zależy od tego, jakie liczby znajdują się w pudełkach.

**Przejdź do nowej linii** sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.



Witajcie!  
Podajcie liczby.

9

11



Pamiętaj, do którego pudełka została wczytana liczba. Jeśli liczba wczytana została do pudełka C i chcesz by Bajtek wypisał wczytaną liczbę, to używając instrukcji „Wypisz pudełko”, musisz polecić Bajtkowi wypisać wartość pudełka C.

Wczytaj do C

Wypisz pudełko C

Jeśli chcesz wczytać kilka liczb, musisz skorzystać z różnych pudełek (np. A i B). Wczytanie dwóch liczb do jednego pudełka (np. do A) spowoduje, że najpierw trafi tam pierwsza liczba, a następnie, po wczytaniu drugiej, zostanie ona usunięta.

Wczytaj do A

Wczytaj do A

Wczytaj do A

Wczytaj do B

## Przykład 1.

Napisz program, który wypisze na konsoli liczbę 2.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Użytkownik nie podaje żadnych danych na wejściu. Komputer za każdym razem wypisze na wyjściu ten sam komunikat, to jest liczbę 2.	2

Przy tworzeniu tego programu powinniśmy zadać sobie następujące pytania:

1. Czy przy każdym uruchomieniu programu mamy wypisać to samo, niezależnie od tego, co podaliśmy na wejściu do programu?
2. Czy mamy wypisać różne wyniki w zależności od tego, jakie było wejście?

Gdyby odpowiedź na pierwsze pytanie brzmiała „tak”, to użylibyśmy instrukcji `Wypisz napis`.

Gdyby odpowiedź na drugie pytanie brzmiała „tak”, to użylibyśmy instrukcji `Wypisz pudełko`.

W naszym przykładzie za każdym razem chcemy wypisać to samo. Używamy więc instrukcji `Wypisz napis 2`.

## Przykład 2.

Napisz program, który wczyta liczbę, wypisze wczytaną liczbę, a następnie w oddzielnym wierszu wypisze 37.

Dzielimy zadanie na kroki, które kolejno musimy wykonać.

Wczytaj liczbę | wypisz tę liczbę | w oddzielnym wierszu | wypisz 37.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

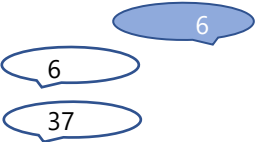
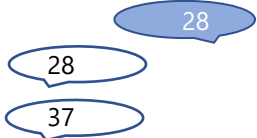
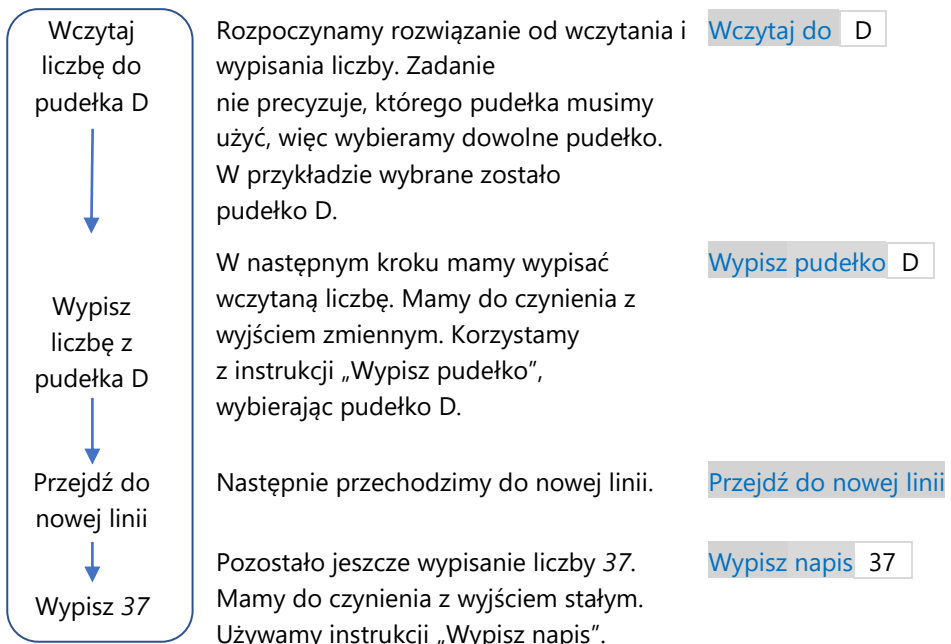
Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 6 program powinien w osobnych wierszach wypisać na wyjściu 6 oraz 37.	
Dla podanej na wejściu liczby 28 program powinien w osobnych wierszach wypisać na wyjściu 28 oraz 37.	

Diagram rozwiązania i jego implementacja.



## Zadania InstaKod Rozdział 2. Wyjście stałe i zmienne

### Poziom 1

- IV 2.1 Napisz program, który wczyta liczbę, wypisze literę *a* oraz wczytaną liczbę.
- IV 2.2 Napisz program, który wczyta liczbę, wypisze liczbę 2, a następnie w oddzielnym wierszu wczytaną liczbę.
- IV 2.3 Napisz program, który wczyta liczbę, wypisze wczytaną liczbę, przejdzie do nowej linii i wypisze 7.

### Poziom 2

- IV 2.4 Napisz program, który wczyta liczbę oraz wypisze w osobnych wierszach trzykrotnie wczytaną liczbę.
- IV 2.5 Napisz program, który wczyta liczbę i wypisze na konsoli wczytaną liczbę oraz literę *x*. Następnie przejdzie do nowej linii i dwukrotnie wypisze wczytaną liczbę.
- IV 2.6 Napisz program, który wczyta liczbę i w osobnych wierszach wypisze wczytaną liczbę, liczbę 7 oraz ponownie wczytaną liczbę.

### Poziom 3

- IV 2.7 Napisz program, który wczyta liczbę i w osobnych wierszach wypisze liczbę 3, wczytaną liczbę oraz liczbę 3.
- IV 2.8 Napisz program, który wczyta dwie liczby i w osobnych wierszach wypisze dwukrotnie pierwszą wczytaną liczbę, liczbę 5 oraz drugą wczytaną liczbę.
- IV 2.9 Napisz program, który wczyta dwie liczby i w osobnych wierszach wypisze liczbę 7, dwukrotnie drugą wczytaną liczbę, liczbę 7 oraz wczytane liczby w kolejności wczytania.

## Zadania dla mistrza

- IV 2.10\* Napisz program, który wczyta cztery liczby i wypisze je w oddzielnych wierszach w kolejności odwrotnej do wczytania. Każda liczba powinna być poprzedzona komunikatem *Liczba:* .
- IV 2.11\* Królik Franek wraz z połową tuzina jego kolegów zasadził w swoim ogrodzie marchewki. Postanowił zaznaczyć teren swojego ogrodu, żeby nikt niepowołany tam nie wszedł. Królik nie zna jednak jeszcze za dobrze liter, więc zamiast swojego imienia na tabliczce przed ogrodzeniem napisał xxx. Napisz program, który wczyta, ile marchewek zasadził w swoim ogrodzie Franek i w oddzielnych wierszach wypisze co napisał królik na tabliczce, liczbę zasadzonych marchewek oraz ilu kolegów pomagało mu w pracy.



## InstaKod 3

### Operacje arytmetyczne 1

#### Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę.

Rolą programisty jest napisanie programu, który z podanych przez użytkownika danych wejściowych wygeneruje oczekiwane dane wyjściowe.

Poznałeś/poznałaś 4 podstawowe instrukcje języka Assembly, które pozwolą ci wczytywać i wypisywać dane oraz komunikować się z użytkownikiem.

**Wczytaj do** A

jest wejściem danych do programu. Przy każdym uruchomieniu programu prosi użytkownika o podanie liczby i zapisuje ją w pudełku A. Cokolwiek było wcześniej zapisane w pudełku A, zostanie usunięte i zastąpione wczytaną wartością.

**Wypisz pudełko** C

wypisuje na konsoli liczbę znajdującą się w pudełku C. Jest wyjściem danych z programu. Wypisana wartość może być zmienna i zależy od tego, co znajduje się w pudełku C w chwili uruchomienia instrukcji **Wypisz pudełko** C.

**Wypisz napis** Ala 2#

wypisuje na konsoli tekst z ramki. Umożliwia programiście wyświetlenie na konsoli stale tego samego wyjścia. Napis może być pojedynczą literą, fragmentem tekstu, cyfrą, ciągiem cyfr lub symbolem.

**Przejdź do nowej linii**

sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.

Poznamy teraz nowe instrukcje służące do wykonywania operacji arytmetycznych na pudełkach.

**Zwiększ A o 8** powoduje dodanie do wartości znajdującej się w pudełku A liczby 8 i zapisanie wyniku w pudełku A. Jeśli w A było 7, to po zwiększeniu o 8 w pudełku A zapisane zostanie 15.

**Zmniejsz B o 1** powoduje odjęcie od wartości aktualnie znajdującej się w pudełku B liczby 1 i zapisanie wyniku w pudełku B. Jeśli w B było 3, to po zmniejszeniu o 1 w B zapisane zostanie 2.

Inną przydatną instrukcją jest „Ustaw”.

**Ustaw A na 8** zapisuje w pudełku A liczbę 8. Liczba, która wcześniej znajdowała się w pudełku A, zostaje usunięta. Mówimy, że przypisaliśmy liczbę 8 do pudełka A.

Co zrobi Bajtek, gdy dostanie instrukcję „Ustaw A na 8”, a co gdy „Zwiększ A o 8”?

Otrzymując instrukcję **Ustaw A na 8**, Bajtek usunie liczbę, która była w pudełku A i wstawi tam nową liczbę, czyli 8.

Gdy otrzyma instrukcję **Zwiększ A o 8**, sprawdzi, jaka liczba jest w pudełku A, doda do niej 8 i wynik włoży do pudełka A, wyrzucając poprzednią liczbę.

Gdy w pudełku A Bajtek miał liczbę 0, to po wykonaniu obu instrukcji znajdzie się tam liczba 8.

Jeśli w A była liczba różna od 0, np. 53, to po wykonaniu instrukcji **Ustaw A na 8** liczba 53 zamieniona zostanie na 8. Po wykonaniu instrukcji **Zwiększ A o 8** liczba 53 zamieniona zostanie na 61.

The image shows two screenshots of a programming environment. Each screenshot consists of three panels: 'Pudełka' (Variables), 'Edytor' (Editor), and 'Konsola' (Console).

**Top Screenshot:**

- Pudełka:** Variable A is 53, B is 0, C is 0, and D is 0. A red slash is over the value 53 in A.
- Edytor:** The instruction 'Ustaw A na 8' is selected in the editor.
- Konsola:** The value 53 is displayed in a blue speech bubble.

**Bottom Screenshot:**

- Pudełka:** Variable A is 61, B is 0, C is 0, and D is 0. A red slash is over the value 61 in A.
- Edytor:** The instruction 'Zwiększ A o 8' is selected in the editor.
- Konsola:** The value 53 is displayed in a blue speech bubble.


## Przykład 1.

Napisz program, który wczyta liczbę do pudełka D, zwiększy ją o 6 i wypisze wynik.

Dzielimy zadanie na kroki, które kolejno musimy wykonać.

Wczytaj liczbę do D | zwiększ ją o 6 | wypisz wynik.

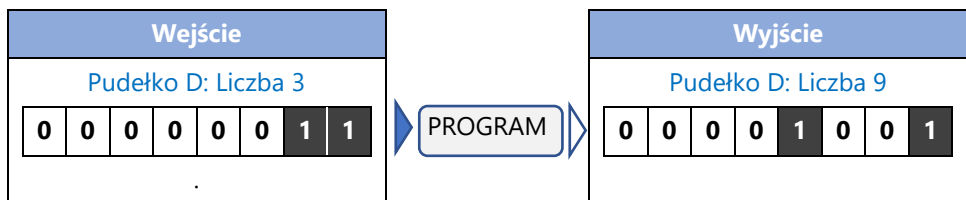
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 3 program powinien wypisać na wyjściu 9 (ponieważ $3 + 6 = 9$ ).	

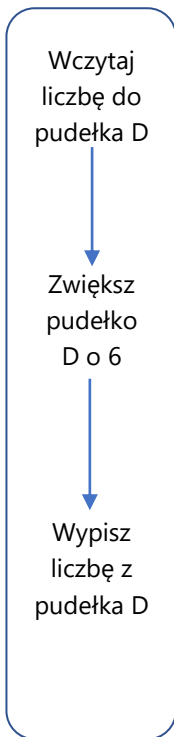
W jaki sposób zapisane są dane wejściowe i dane wyjściowe? Oczywiście za pomocą bitów w komórkach pamięci.

Na wejściu podana została liczba 3, która zapisana została w pudełku D jako 00000011.

Na wyjściu pojawiła się liczba 9, która zapisana była w pudełku D jako 00001001.



## Diagram rozwiązania i jego implementacja.



Najpierw musimy wczytać dane. Używamy instrukcji „Wczytaj”. Zadanie precyzuje pudełko, do którego mamy wczytać liczbę, więc wybieramy pudełko D.

Wczytaj do D

Po wczytaniu danych możemy przejść do obliczeń arytmetycznych. Mamy zwiększyć wczytaną liczbę o 6. Wybieramy instrukcję „Zwiększ” oraz odpowiednie pudełko i wpisujemy wartość, o którą zamierzamy zwiększyć wczytaną liczbę. Pomyśl, co by się stało, gdybyśmy zamiast pudełka D zwiększyli pudełko A.

Zwiększ D o 6

Wypisujemy zwiększoną liczbę na konsolę. Wybieramy instrukcję „Wypisz pudełko” oraz odpowiednie pudełko. Czy wiesz, dlaczego nie wypisujemy teraz pudełka A? Jeśli nie, spróbuj przetestować taki program i zobacz, co się stanie.

Wypisz pudełko D

## Przykład 2.

Napisz program, który wczyta liczbę do pudełka C, zmniejszy tę liczbę o 3 i wypisze wynik. Następnie przejdzie do nowej linii i wypisze poprzedni wynik powiększony o 5.

Dzielimy zadanie na kroki, które kolejno musimy wykonać.

Wczytaj liczbę do pudełka C | zmniejsz tę liczbę o 3 | wypisz wynik | przejdź do nowej linii | wypisz poprzedni wynik powiększony o 5.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 8 program powinien wypisać na wyjściu 5 (ponieważ $8 - 3 = 5$ ) oraz 10 (ponieważ $5 + 5 = 10$ ).	<pre>8 5 10</pre>

W jaki sposób zapisane są liczbowe dane wejściowe i dane wyjściowe?  
Oczywiście za pomocą bitów w komórkach pamięci.

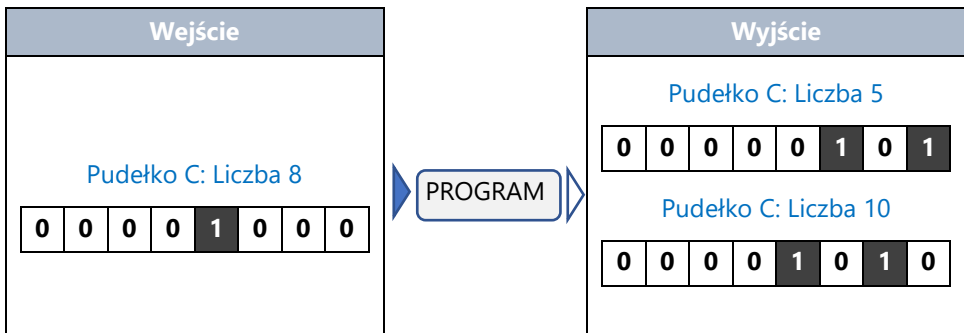
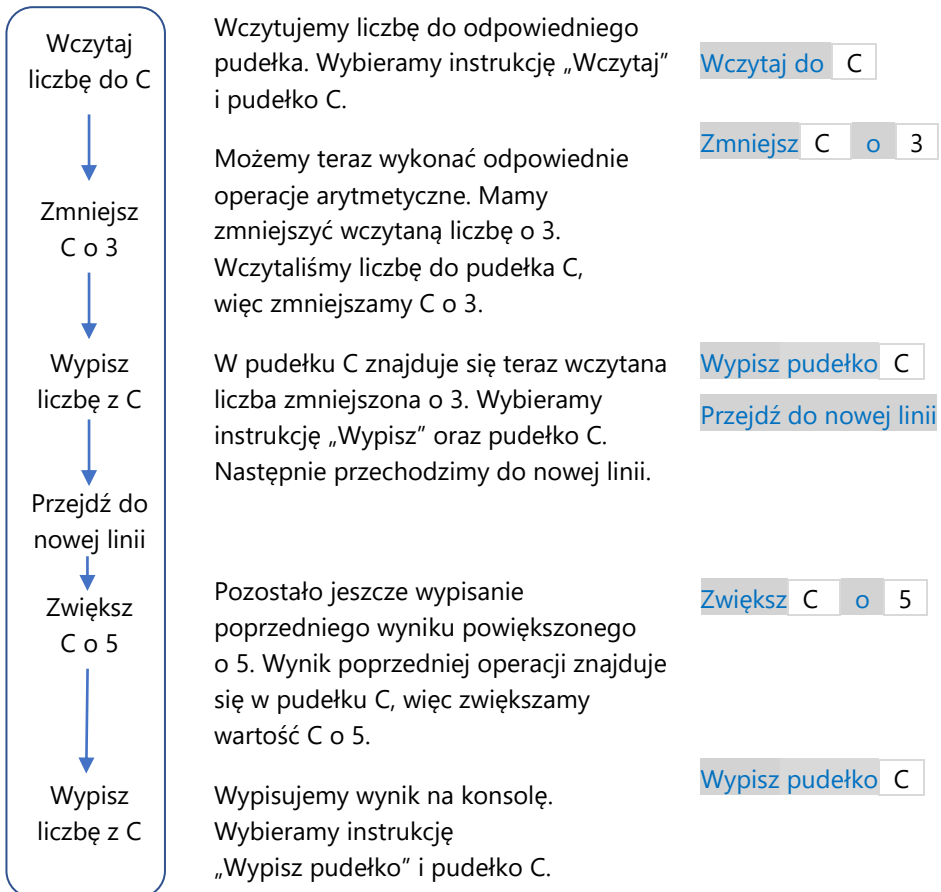


Diagram rozwiązania i jego implementacja.



## Zadania InstaKod 3. Operacje arytmetyczne 1

### Poziom 1

- IV 3.1 Napisz program, który wczyta liczbę do pudełka B, zmniejszy ją o 2 i wypisze nową wartość pudełka B.
- IV 3.2 Napisz program, który wczyta liczbę do pudełka A, a następnie doda do niej 5 i wypisze wynik.
- IV 3.3 Napisz program, który wczyta liczbę do pudełka A, zmniejszy ją o 8 i w oddzielnych wierszach wypisze nową wartość pudełka A oraz liczbę 3.

### Poziom 2

- IV 3.4 Napisz program, który wczyta liczbę do pudełka C i wypisze wartość pudełka C. Następnie zwiększy ją o 10 i w nowym wierszu wypisze końcową wartość C.
- IV 3.5 Napisz program, który wczyta liczbę do pudełka A, wypisze wartość pudełka A, zmniejszy A o 1 i wypisze wynik w nowej linii.
- IV 3.6 Napisz program, który wczyta liczbę do pudełka A, a następnie doda do niej 3 i w oddzielnych wierszach wypisze wynik oraz liczbę 5.

### Poziom 3

- IV 3.7 Napisz program, który wczyta liczbę do pudełka A, zwiększy liczbę w pudełku A o 5 i wypisze różnicę uzyskanej sumy i liczby 3. Następnie w nowym wierszu wypisze liczbę 3.
- IV 3.8 Napisz program, który wczyta liczbę do pudełka D i ją wypisze. Następnie w nowej linii wypisze dwukrotność wczytanej liczby.
- IV 3.9 Napisz program, który wczyta liczbę do pudełka C i wypisze liczbę 2. Następnie w nowej linii wypisze wartość pudełka C pomniejszoną o 2.

### Zadania dla mistrza

- IV 3.10\* Napisz program, który wczyta liczbę do pudełka D, zwiększy ją o 3 i wypisze wynik. Następnie przejdzie do nowej linii i dwukrotnie wypisze wartość pudełka D.

IV 3.11\* Mama wysłała Anię na zakupy. Ania miała kupić jajka oraz sześć pomarańczy. Droga z domu do sklepu była wyboista. Dziewczynka, jadąc na rowerze, straciła równowagę, przez co potłukły się dwa jajka. Gdy mama spytała Anię, ile jajek i pomarańczy przyniosła do domu, Ania dwukrotnie podała liczbę niepotłuczonych jajek, a następnie liczbę pomarańczy. Napisz program, który wczyta do pudełka C, ile jajek kupiła w sklepie Ania i w oddzielnych wierszach wypisze, co odpowiedziała mamie.

## InstaKod 4

### Operacje arytmetyczne 2

Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę. Krok dalej.

Dotychczas poznaliśmy następujące pojęcia i instrukcje:

#### Wejście programu

**Wczytaj do** B jest wejściem danych do programu. Przy każdym uruchomieniu programu instrukcja poprosi użytkownika o podanie liczby. Wczytana liczba zostaje zapisana w pudełku B. Liczba, która wcześniej znajdowała się w B, zostanie usunięta.

#### Wyjście programu, stałe i zmienne

Wyjście to dane wypisane przez program w celu komunikacji z użytkownikiem.

W Assembly mamy trzy instrukcje wyjścia. Są to:

**Wypisz pudełko** C wypisuje na konsoli liczbę znajdującą się w pudełku C. Wypisana wartość może być zmienna i zależy od tego, jaka liczba aktualnie znajduje się w pudełku C.

**Wypisz napis** Ala 45# wypisuje na konsoli tekst z ramki. Służy programiście do wyświetlania komunikatu dla użytkownika programu. Wypisywany komunikat jest stały i nie zależy od wczytanych danych.

**Przejdź do nowej linii** sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.

#### Operacje arytmetyczne

**Zwiększ** D o 5 oblicza sumę liczby znajdującej się w pudełku D oraz liczby 5 i zapisuje wynik w pudełku D.



Zmniejsz C o 2 oblicza różnicę liczby znajdującej się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

Tworząc programy, które wykonują operacje arytmetyczne na wczytanych liczbach, pamiętaj, do którego pudełka wczytałeś/wczytałaś liczbę.


## Przykład 1.

Napisz program, który wczyta liczbę, doda do niej 3 i wypisze wynik.

Dzielimy zadanie na poszczególne kroki:

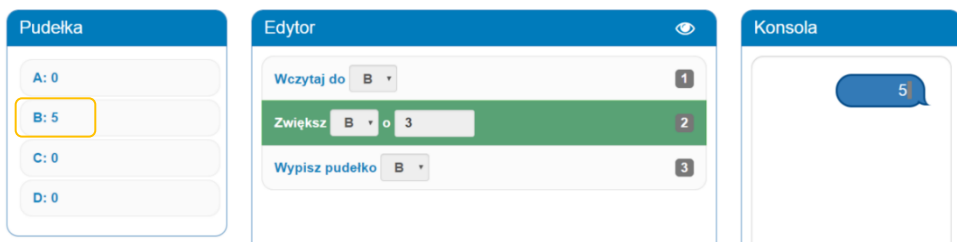
Wczytaj liczbę | dodaj do niej 3 | wypisz wynik.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

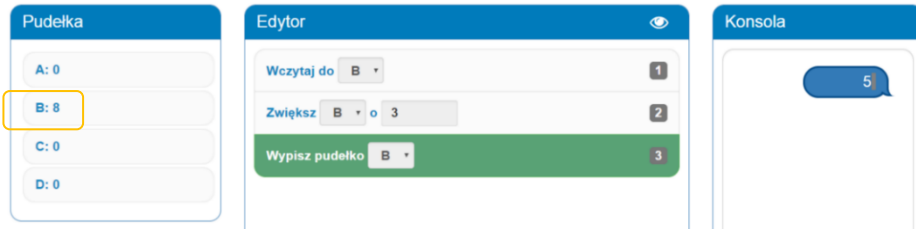
Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 5 program powinien wypisać na wyjściu 8 (ponieważ $5 + 3 = 8$ ).	

Na początku musimy zdecydować się na pudełko, do którego wczytamy liczbę. Wybierzmy pudełko B. Następnym krokiem jest zwiększenie pudełka, do którego wczytaliśmy liczbę o 3. Na koniec wypisujemy nową wartość pudełka B.

Uruchommy nasz program. Zróbmy pierwszy krok. Wczytajmy liczbę 5. Wczytana liczba trafi do pudełka B.



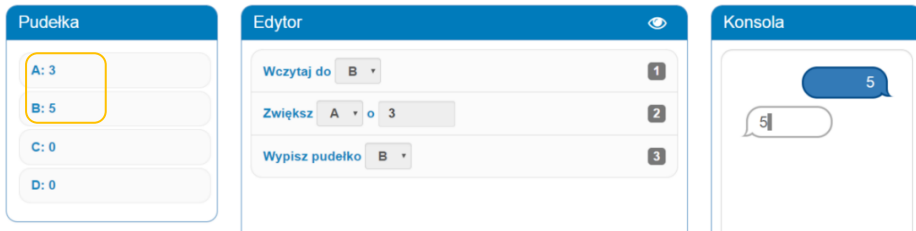
Zróbmy drugi krok. Liczba w pudełku B zostanie zwiększona o 3, więc przyjmie wartość 8.



W trzecim kroku na konsoli zostanie wypisana wartość pudełka B.

### Przykład rozwiązania z błędami

Co się stanie, gdy wczytamy liczbę do B, a zwiększymy liczbę w pudełku A? Oczywiście liczba w B nie zmieni się, za to liczba w A (jeśli nie wykonaliśmy w pudełku żadnej operacji, jest tam 0) zostanie powiększona o 3.



Takie rozwiązanie nie przejdzie testów.

Dla wczytanej liczby 1 oczekiwany wynik to 4, a nasz wynik to 1.

Dla wczytanej liczby 5 oczekiwany wynik to 8, a nasz wynik to 5.

Dla wczytanej liczby 2 oczekiwany wynik to 5, a nasz wynik to 2.

Testowanie programu					
Twój program nie przeszedł poniższych testów					
Rezultat	Twoja odpowiedź	Przewidywana odpowiedź	Twoje pudełka	Przewidywane pudełka	Kroków
Zła odpowiedź			A: 1 B: 3 C: 0 D: 0		3
Zła odpowiedź			A: 5 B: 3 C: 0 D: 0		3
Zła odpowiedź			A: 2 B: 3 C: 0 D: 0		3

Prawidłowo wszystkie instrukcje musimy wykonać na tym samym pudełku. Jeśli wczytamy liczbę do pudełka B, to musimy zwiększyć liczbę w pudełku B i wypisać liczbę z pudełka B.

## Przykład 2.

Napisz program, który wczyta liczbę, odejmie od niej 10, doda 20, wypisze napis *a3b* i wynik działania, a następnie w nowej linii 77.

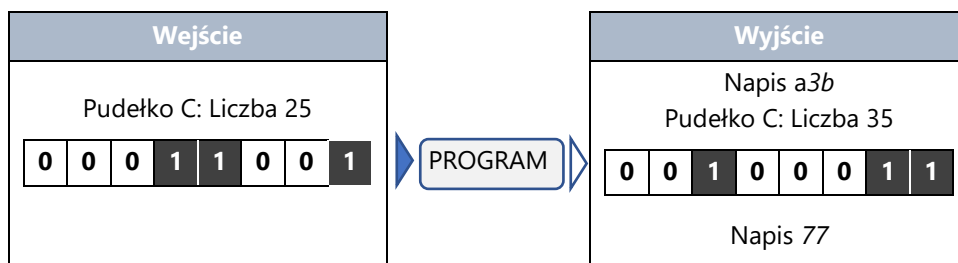
Dzielimy zadanie na poszczególne kroki:

Wczytaj liczbę | odejmij od niej 10 | dodaj 20 | wypisz napis *a3b* | wypisz wynik (wczytana liczba - 10 + 20) | przejdź do nowej linii | wypisz 77.

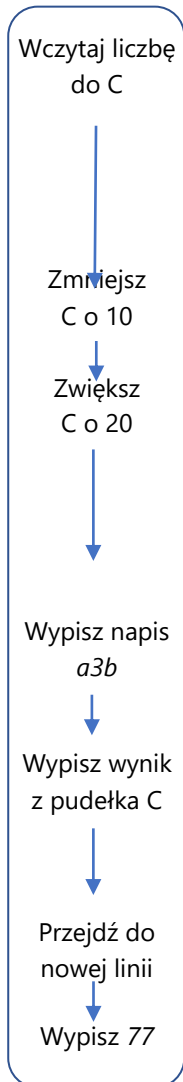
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 25 program powinien wypisać na wyjściu <i>a3b</i> , liczbę 35 (ponieważ $25 - 10 + 20 = 35$ ) i w nowej linii 77.	<pre> 25 a3b35 77 </pre>
Dla podanej na wejściu liczby 12 program powinien wypisać na wyjściu <i>a3b</i> , liczbę 22 (ponieważ $12 - 10 + 20 = 22$ ) i w nowej linii 77.	<pre> 12 a3b22 77 </pre>

W jaki sposób zapisane są liczbowe dane wejściowe i dane wyjściowe? Oczywiście za pomocą bitów w komórkach pamięci poszczególnych pudełek.



## Diagram rozwiązania i jego implementacja.



Zaczynamy od wczytania liczby. Wybieramy instrukcję „Wczytaj” i dowolne pudełko. W przykładzie wybrane zostało pudełko C. Pamiętajmy, że wczytaną liczbę zapisałiśmy w C.

Wczytaj do C

Możemy teraz wykonać odpowiednie operacje arytmetyczne. Mamy zmniejszyć wczytaną liczbę o 10, a następnie zwiększyć ją o 20.

Zmniejsz C o 10  
Zwiększ C o 20

Wybieramy najpierw instrukcję `Zmniejsz C o 10`, a potem `Zwiększ C o 20`. Wynik zostaje zapisany w pudełku C. Pomyśl, czy zmiana kolejności wykonywanych instrukcji wpłynęłaby na wynik.

Wypiszmy teraz  $a3b$ . Do wypisywania stale tego samego napisu służy instrukcja „Wypisz napis”.

Wypisz napis a3b

Następnie wypisujemy wynik naszych obliczeń, który znajduje się w pudełku C. Wybieramy instrukcję „Wypisz pudełko” oraz pudełko C.

Wypisz pudełko C

Do wypisywania stale tego samego komunikatu służy instrukcja wyjścia „Wypisz napis”.

Przejdź do nowej linii

Wypisz napis 77

## Zadania InstaKod 4. Operacje arytmetyczne 2.

### Poziom 1

- IV 4.1 Napisz program, który wczyta liczbę, zwiększy wczytaną liczbę o 2 i wypisze wynik.
- IV 4.2 Napisz program, który wczyta liczbę, wypisze wczytaną liczbę, następnie zwiększy ją o 15 i w nowym wierszu wypisze wynik.
- IV 4.3 Napisz program, który wczyta liczbę i w oddzielnych wierszach wypisze 3 oraz wczytaną liczbę pomniejszoną o 2.

### Poziom 2

- IV 4.4 Napisz program, który wczyta liczbę, zmniejszy ją o 5, zwiększy o 6 i dwukrotnie wypisze wynik.
- IV 4.5 Napisz program, który wczyta liczbę i wypisze tę liczbę powiększoną o 2, a następnie w nowej linii wypisze 3.
- IV 4.6 Napisz program, który wczyta liczbę, wypisze napis  $a3h$ , następnie odejmie od wczytanej liczby 1 i wypisze wynik.

### Poziom 3

- IV 4.7 Napisz program, który wczyta liczbę, zwiększy ją o 1, zmniejszy o 2 i wypisze wynik. Następnie w nowej linii wypisze liczbę 8.
- IV 4.8 Napisz program, który wczyta liczbę, wypisze sumę wczytanej liczby i liczby 10. Następnie w nowej linii wypisze różnicę obliczonej sumy oraz liczby 3.
- IV 4.9 Napisz program, który wczyta liczbę i wypisze liczbę 11. Następnie w nowej linii wypisze różnicę liczby 30 oraz wczytanej liczby.

### Zadania dla mistrza

- IV 4.10\* Napisz program, który wczyta cztery liczby i wypisze je w oddzielnych wierszach powiększone o kolejne wielokrotności liczby 2 (zaczynając od 2).

IV 4.11\* Stasiek jechał na rowerze do szkoły. Jak zwykle, rozpędził się do pewnej prędkości i jechał w tym tempie dalej. Zobaczył z daleka swojego kolegę i by go dogonić przyspieszył zwiększając swoją prędkość o 5 kilometrów na godzinę. Gdy go dogonił, zmniejszył prędkość o 3 kilometry na godzinę, by jechać równo z kolegą. Napisz program, który wczyta prędkość, do której rozpędził się zanim zobaczył kolegę i wypisze na konsoli w oddzielnych wierszach z jaką prędkością dogonił kolegę oraz z jaką prędkością chłopcy jechali razem.

## InstaKod 5

### Operacje arytmetyczne 3

#### Zwiększanie i zmniejszanie wartości pudełka o wartość wybranego pudełka.

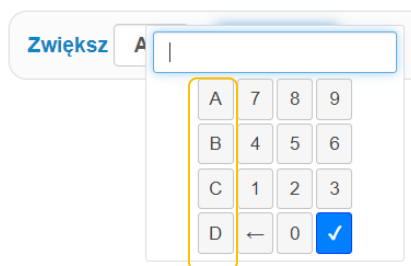
Wiesz już, jak zwiększyć lub zmniejszyć liczbę zapisaną w pudełku o wybraną liczbę.

#### Operacje arytmetyczne

**Zwiększ** D o 5 oblicza sumę liczby znajdującej się w pudełku D oraz liczby 5 i zapisuje wynik w pudełku D.

**Zmniejsz** C o 2 oblicza różnicę liczby znajdującej się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

Zwróć uwagę, że przy uzupełnianiu wartości drugiego pola instrukcji „Zwiększ” oraz „Zmniejsz” pojawia się klawiatura zawierająca nie tylko cyfry, ale też nazwy pudełek: A, B, C i D.



Możemy więc zwiększyć liczbę w pudełku A nie tylko o liczbę 5, 7 czy 102, ale też o wartość innego pudełka, np. pudełka B.

Gdy komputer Bajtek otrzyma instrukcję **Zwiększ** A o B, to zwiększy liczbę aktualnie znajdującą się w pudełku A o wartość pudełka B.



Gdy Bajtek otrzyma instrukcję `Zmniejsz C o D`, to zmniejszy liczbę z pudełka C o liczbę z pudełka D.



### Operacje arytmetyczne

`Zwiększ D o 5` oblicza sumę liczby znajdującej się w pudełku D oraz liczby 5 i zapisuje wynik w pudełku D.

`Zmniejsz C o 2` oblicza różnicę liczby znajdującej się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

`Zwiększ A o C` oblicza sumę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku C i zapisuje wynik w pudełku A.

`Zmniejsz A o C` oblicza różnicę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku C i zapisuje wynik w pudełku A.

### Przykład 1.

Napisz program, który wczyta liczby do pudełek A i B. Następnie w oddzielnych wierszach wypisze A zwiększone o 1 oraz B.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 1 i 3 program powinien na wyjściu wypisać 2 (ponieważ $1 + 1 = 2$ ) oraz 3.	
Dla podanych na wejściu liczb 14 i 26 program powinien wypisać na wyjściu 15 (ponieważ $14 + 1 = 15$ ) oraz 26.	



W jaki sposób zapisane są dane wejściowe i dane wyjściowe?

Oczywiście za pomocą bitów w komórkach pamięci poszczególnych pudełek.

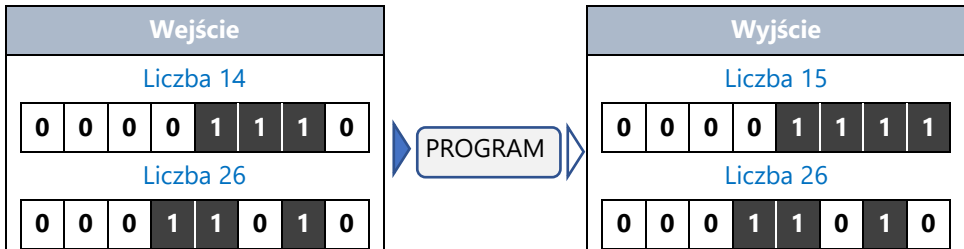
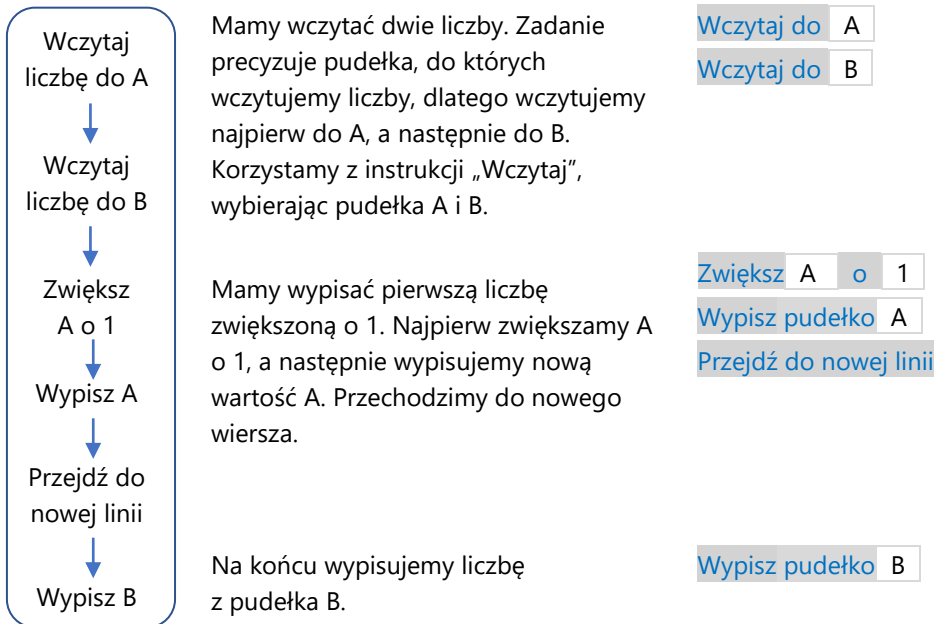


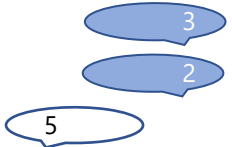
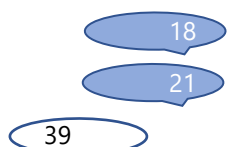
Diagram rozwiązania i jego implementacja.



## Przykład 2.

Napisz program, który wczyta liczby do pudełek A i B oraz wypisze ich sumę.

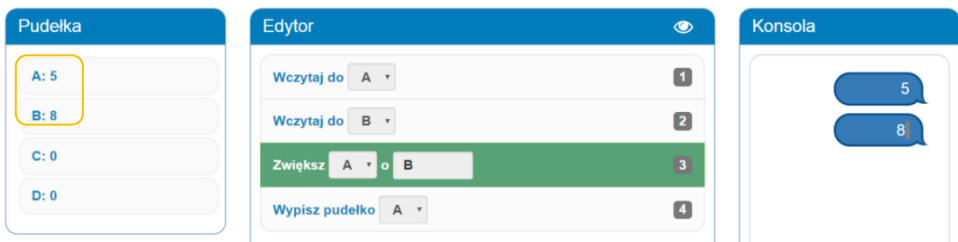
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 3 i 2 program powinien wypisać na wyjściu 5 (ponieważ $3 + 2 = 5$ ).	
Dla podanych na wejściu liczb 18 i 21 program powinien wypisać na wyjściu 39 (ponieważ $18 + 21 = 39$ ).	

Aby obliczyć sumę dwóch wczytanych liczb, musimy zwiększyć jedną z nich o drugą. Załóżmy, że wczytamy liczby do pudełek A i B. Zwróćmy uwagę, że możemy zrobić to na dwa sposoby.

- $A + B$  – gdy zwiększymy A o B, to wynik zostanie zapisany w A.
- $B + A$  – gdy zwiększymy B o A, to wynik zostanie zapisany w B.

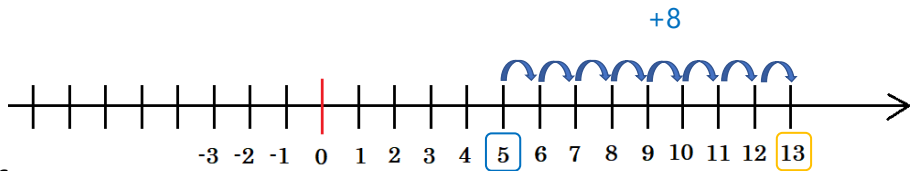
Nasz program może wyglądać jak poniżej. Uruchommy go, wczytując liczby 5 i 8. Pierwsza liczba trafi do pudełka A, a druga do pudełka B.



The screenshot shows a program interface with three main panels:

- Pudełka:** A panel with four input fields labeled A, B, C, and D. Fields A and B are highlighted with a yellow border and contain the values 5 and 8 respectively. Fields C and D contain 0.
- Edytor:** A panel with four numbered steps:
  1. Wczytaj do A (with a dropdown menu)
  2. Wczytaj do B (with a dropdown menu)
  3. Zwiększ A o B (highlighted in green)
  4. Wypisz pudełko A (with a dropdown menu)
- Konsola:** A panel showing the output of the program, with two blue speech bubbles containing the numbers 5 and 8.

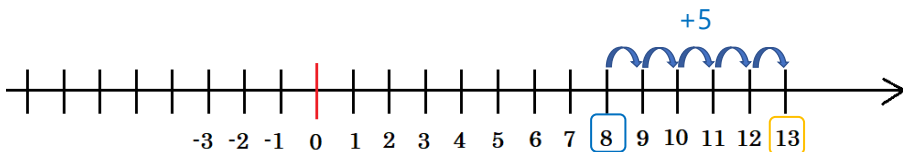
Zilustrujemy dodawanie liczb w pudełkach A i B za pomocą instrukcji **Zwiększ A o B** na osi liczbowej dla wczytanych liczb  $A = 5$  i  $B = 8$ . Zaczynamy w 5 i przesuwamy się o 8 kresek w prawo do liczby 13.



Nasza instrukcja **Zwiększ A o B** liczba w pudełku A zostanie zaktualizowana do 13 (ponieważ  $5 + 8 = 13$ ). Liczba w pudełku B nie ulegnie zmianie.

Pudełka	Edytor	Konsola
A: 13 B: 8 C: 0 D: 0	Wczytaj do A 1 Wczytaj do B 2 Zwiększ A o B 3 Wypisz pudełko A 4	5 8 13

Zilustrujemy dodawanie liczb w pudełkach A i B na osi liczbowej dla wczytanych liczb  $A = 5$  i  $B = 8$  za pomocą operacji **Zwiększ B o A**. Zaczynamy w 8 i przesuwamy się o 5 kresek w prawo do liczby 13.



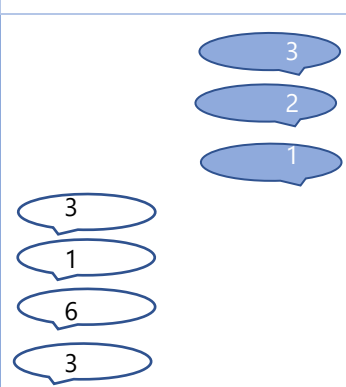
Poniższy program zapisuje sumę  $B + A$  w pudełku B. Po wykonaniu instrukcji **Zwiększ B o A** liczba w pudełku B zostanie zaktualizowana do 13 (ponieważ  $5 + 8 = 13$ ). Tym razem liczba w pudełku A nie ulegnie zmianie.

Pudełka	Edytor	Konsola
A: 5 B: 13 C: 0 D: 0	Wczytaj do A 1 Wczytaj do B 2 Zwiększ B o A 3 Wypisz pudełko B 4	5 8 13

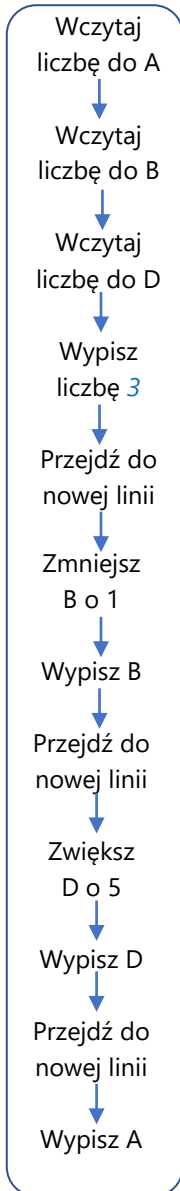
### Przykład 3.

Napisz program, który wczyta liczby do pudełek A, B, D i wypisze w oddzielnych wierszach liczbę 3 oraz B zmniejszone o 1. Następnie program przejdzie do nowej linii i w oddzielnych wierszach wypisze liczbę z pudełka D zwiększonego o 5 oraz pudełko A.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 3, 2 i 1 program powinien wypisać na wyjściu w oddzielnych wierszach 3 i 1 (ponieważ $2 - 1 = 1$ ) oraz również w oddzielnych wierszach 6 (ponieważ $1 + 5 = 6$ ) i 3.	 <p>The diagram shows a vertical line representing the console. On the right side, three blue speech bubbles point towards the line, containing the numbers 3, 2, and 1 from top to bottom. On the left side, four white speech bubbles point away from the line, containing the numbers 3, 1, 6, and 3 from top to bottom.</p>

## Diagram rozwiązania i jego implementacja.



Wczytujemy liczby do pudełek A, B i D.

Czy wiesz, dlaczego istotne jest zachowanie właściwej kolejności?

Co by się stało, gdybyśmy wczytali liczby w kolejności najpierw do D, potem do A, a na końcu do B?

Liczba 3 nie zależy w żaden sposób od wczytanych danych i jest w naszym zadaniu stałą. Możemy od razu wypisać ją na konsoli, używając instrukcji „Wypisz napis”. Przechodzimy do nowej linii.

Mamy wypisać B zmniejszone o 1. Oczywiście najpierw musimy zmniejszyć liczbę, a dopiero potem ją wypisać. Zmniejszamy B o 1 i wypisujemy B. Następnie przechodzimy do nowej linii.

Mamy wypisać liczbę z pudełka D zwiększoną o 5. Najpierw obliczamy sumę  $D + 5$ , a następnie wypisujemy wynik na konsolę. Przechodzimy do nowej linii.

Na końcu wypisujemy pudełko A.

Wczytaj do A

Wczytaj do B

Wczytaj do D

Wypisz napis 3

Przejdź do nowej linii

Zmniejsz B o 1

Wypisz pudełko B

Przejdź do nowej linii

Zwiększ D o 5

Wypisz pudełko D

Przejdź do nowej linii

Wypisz pudełko A

## Zadania InstaKod 5. Operacje arytmetyczne 3.

### Poziom 1

- IV 5.1 Napisz program, który wczyta liczby do pudełek A i B, następnie w oddzielnych wierszach wypisze liczbę wczytaną do B oraz liczbę wczytaną do A pomniejszoną o 1.
- IV 5.2 Napisz program, który wczyta liczby do pudełek B i C, a następnie wypisze ich sumę.
- IV 5.3 Napisz program, który wczyta liczby do pudełek B i D, następnie wypisze D, przejdzie do nowej linii i wypisze B zmniejszone o D.

### Poziom 2

- IV 5.4 Napisz program, który wczyta liczbę do pudełka C, zwiększy C o 3 i wypisze nową wartość pudełka C. Następnie wczyta liczbę do pudełka A i w nowym wierszu wypisze wczytaną wartość.
- IV 5.5 Napisz program, który wczyta liczbę do pudełka B, w oddzielnych wierszach wypisze tę liczbę powiększoną o 3 oraz liczbę 5. Następnie wczyta liczbę do pudełka C i w nowym wierszu wypisze C.
- IV 5.6 Napisz program, który wczyta liczby do pudełek A, B i D, w oddzielnych wierszach wypisze B zwiększone o 2, D zmniejszone o 1 oraz A zwiększone o 9.

### Poziom 3

- IV 5.7 Napisz program, który wczyta liczby do pudełek A, B i C oraz wypisze ich sumę pomniejszoną o 6.
- IV 5.8 Napisz program, który wczyta liczby do pudełek C, A i D. Następnie w osobnych wierszach wypisze sumę liczb z pudełek A i C oraz różnicę liczb z pudełek D i A.
- IV 5.9 Napisz program, który wczyta liczby do pudełek D, C i B. Następnie w osobnych wierszach wypisze różnicę liczb z pudełek C i D oraz obliczoną różnicę powiększoną o liczbę z pudełka B.

### **Zadania dla mistrza**

- IV 5.10\* Napisz program, który wczyta liczby do pudełek D, C, B i A, w oddzielnych wierszach wypisze wczytane liczby pomniejszone o 1 w kolejności odwrotnej do wczytywania oraz sumę wczytanych liczb.
- IV 5.11\* Pan ogrodnik ma w swoim ogrodzie wiele rodzajów kwiatów. Wśród nich są kwiaty żółte, czerwone i fioletowe. Napisz program, który wczyta do pudełek A, B i C, ile było żółtych, czerwonych i fioletowych kwiatów w ogrodzie ogrodnika i wypisze na konsoli, ile kwiatów było razem.

## Instakod 6

### Operacje arytmetyczne 4

Zwiększanie i zmniejszanie wartości pudełka o wartość wybranego pudełka. Krok dalej.

#### Suma i różnica dwóch liczb

**Zwiększ** A  B oblicza sumę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku B i zapisuje wynik w pudełku A.

**Zwiększ** B  A oblicza sumę liczby znajdującej się w pudełku B oraz liczby znajdującej się w pudełku A i zapisuje wynik w pudełku B.

**Zmniejsz** A  B oblicza różnicę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku B i zapisuje wynik w pudełku A.

**Zmniejsz** B  A oblicza różnicę liczby znajdującej się w pudełku B oraz liczby znajdującej się w pudełku A i zapisuje wynik w pudełku B.

#### Przykład 1.

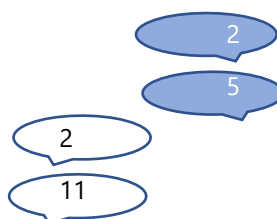
Napisz program, który wczyta dwie liczby i w oddzielnych wierszach wypisze pierwszą wczytaną liczbę oraz drugą wczytaną liczbę zwiększoną o 6.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

#### Wejście i wyjście

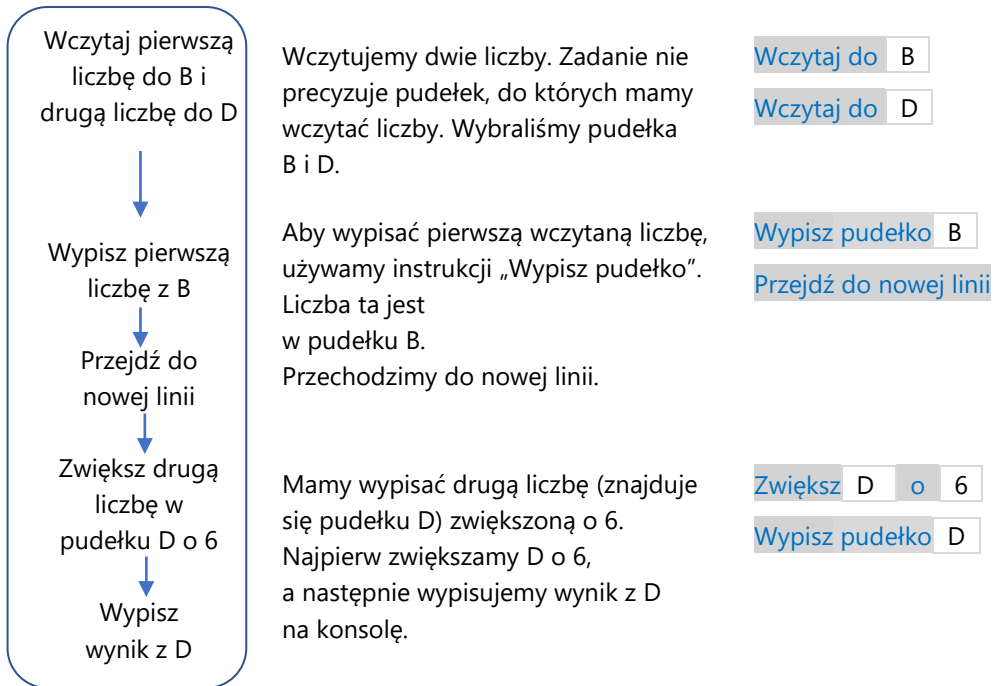
Dla podanych na wejściu liczb 2 i 5 program powinien wypisać na wyjściu w oddzielnych wierszach 2 i 11 (ponieważ  $5 + 6 = 11$ ).

#### Konsola





## Diagram rozwiązania i jego implementacja.



## Przykład 2.

Napisz program, który wczyta dwie liczby i w oddzielnych wierszach wypisze pierwszą oraz różnicę wczytanych liczb.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 11 i 6, program powinien wypisać na wyjściu w oddzielnych wierszach 11 i 5 (ponieważ $11 - 6 = 5$ ).	<p>The terminal screenshot shows the user entering '11' and '6' on separate lines. The program outputs '11' and '5' on separate lines.</p>

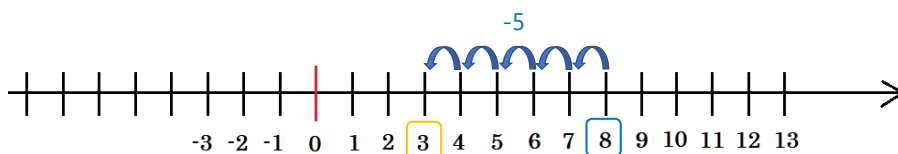
Aby obliczyć różnicę dwóch wczytanych liczb, musimy zmniejszyć jedną z nich o drugą. Załóżmy, że wczytamy liczby do pudełek A i B. Zauważmy, że możemy zrobić to na dwa sposoby:

- **Zmniejsz** A o B – zmniejszymy A o B, czyli pierwszą liczbę o drugą i wynik zostanie zapisany w A,
- **Zmniejsz** B o A – zmniejszymy B o A, czyli drugą liczbę o pierwszą i wynik zostanie zapisany w B.

Nasz program może wyglądać jak poniżej. Uruchommy go, wczytując liczbę 8 do A i 5 do B.

The screenshot shows three panels: 'Pudełka' (Variables), 'Edytor' (Editor), and 'Konsola' (Console). In 'Pudełka', A is 8 and B is 5. In 'Edytor', the code is: 1. Wczytaj do A, 2. Wczytaj do B, 3. Wypisz pudełko A, 4. Przejdź do nowej linii, 5. Zmniejsz A o B, 6. Wypisz pudełko A. The console shows the output '8' and '5'.

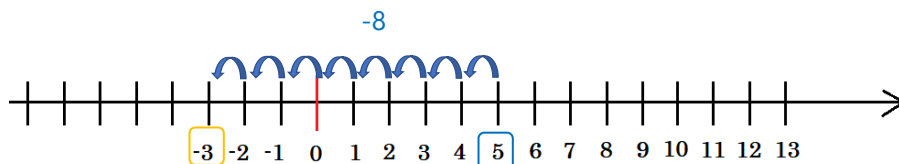
Zilustrujemy operację **A - B** na osi liczbowej dla liczb  $A = 8$  i  $B = 5$ . Zaczynamy w 8 i przesuwamy się o 5 kresek w lewo do liczby 3.



Nasz program zapisuje różnicę  $A - B$  w pudełku A. Po wykonaniu instrukcji **Zmniejsz** A o B liczba w pudełku A zostanie zaktualizowana do 3 (ponieważ  $8 - 5 = 3$ ). Liczba w pudełku B nie zostanie zmieniona.

The screenshot shows the same three panels. In 'Pudełka', A is now 3 and B is still 5. In 'Edytor', the code is the same as before. The console now shows '8', '5', and '3' on separate lines.

Zobaczmy, co by się stało, gdybyśmy policzyli różnicę drugiej i pierwszej liczby, zamiast pierwszej i drugiej. Zilustrujemy operację  $B - A$  na osi liczbowej dla wczytanych liczb  $A=8$  i  $B=5$ . Zaczynamy w 5 i przesuwamy się o 8 kresek w lewo do liczby  $-3$ .



Wykonajmy instrukcję **Zmniejsz B o A** i sprawdźmy, jak zostanie obliczona różnica  $B - A$ . Zauważmy, że teraz w B pojawiła się różnica, to jest  $5 - 8$ , czyli  $-3$ , a w A pozostała wczytana wartość 8.

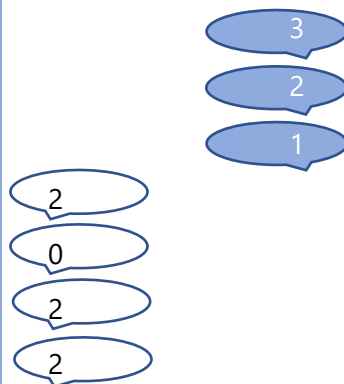
The screenshot shows a programming environment with three panels: 'Pudełka', 'Edytor', and 'Konsola'.  
- **Pudełka:** A: 8, B: -3, C: 0, D: 0. The values for A and B are highlighted with yellow boxes.  
- **Edytor:** A list of instructions:  
1. Wczytaj do A  
2. Wczytaj do B  
3. Wypisz pudełko A  
4. Przejdź do nowej linii  
5. Zmniejsz B o A (highlighted)  
6. Wypisz pudełko B  
- **Konsola:** Shows the output of the instructions: a blue bubble with '8', a blue bubble with '5', and a white bubble with '8' and '-3' on separate lines.

Uwaga: Przy liczeniu różnicy kolejność ma znaczenie!

### Przykład 3.

Napisz program, który wczyta trzy liczby i w oddzielnych wierszach wypisze drugą, pierwszą zmniejszoną o 3, trzecią zwiększoną o 1 i jeszcze raz drugą.

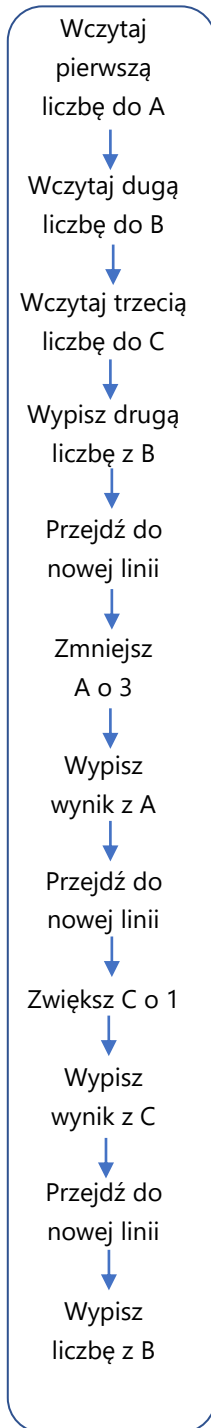
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych przez użytkownika liczb: 3, 2 i 1 program powinien wypisać w oddzielnych wierszach: 2, następnie 0 (ponieważ $3 - 3 = 0$ ), 2 (ponieważ $1 + 1 = 2$ ), i na końcu 2.	

Potrzebujemy wypisać drugą liczbę, pierwszą zmniejszoną o 3, trzecią zwiększoną o 1 i jeszcze raz drugą. Najpierw zaplanujmy, gdzie zapiszemy liczby i jakie wykonamy obliczenia.

Pudełko A: 3	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>	0	0	0	0	0	0	1	1	Pierwsza wczytana liczba
0	0	0	0	0	0	1	1			
Pudełko B: 2	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	0	1	0	Druga wczytana liczba
0	0	0	0	0	0	1	0			
Pudełko C: 1	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	0	0	0	0	1	Trzecia wczytana liczba
0	0	0	0	0	0	0	1			

## Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Mamy wczytać trzy liczby. Zadanie nie precyzuje pudełek, do których wczytujemy, więc wybieramy dowolne pudełka. Ważne jest, by wczytać liczby do trzech różnych pudełek. W przykładzie pierwsza liczba została wczytana do A, druga do B, a trzecia do C.

```
Wczytaj do A
```

```
Wczytaj do B
```

```
Wczytaj do C
```

Mamy najpierw wypisać drugą wczytaną liczbę. Wypisujemy pudełko B. Przechodzimy do nowej linii.

```
Wypisz pudełko B
```

```
Przejdź do nowej linii
```

Następnie mamy wypisać pierwszą wczytaną liczbę pomniejszoną o 3. Zmniejszamy A o 3. Wypisujemy wynik na konsolę. Przechodzimy do nowej linii.

```
Zmniejsz A o 3
```

```
Wypisz pudełko A
```

```
Przejdź do nowej linii
```

Następnie mamy wypisać trzecią wczytaną liczbę powiększoną o 1. Liczymy sumę  $C + 1$  i wypisujemy wynik na konsolę. Przechodzimy do nowej linii.

```
Zwiększ C o 1
```

```
Wypisz pudełko C
```

```
Przejdź do nowej linii
```

Na końcu wypisujemy drugą wczytaną liczbę.

```
Wypisz pudełko B
```

## Zadania. InstaKod Rozdział 6. Operacje arytmetyczne 4.

### Poziom 1

- IV 6.1 Napisz program, który wczyta dwie liczby i wypisze w oddzielnych wierszach drugą zmniejszoną o 2 i pierwszą zwiększoną o 12.
- IV 6.2 Napisz program, który wczyta dwie liczby i wypisze pierwszą zwiększoną o 2. Następnie przejdzie do nowej linii i wypisze drugą pomniejszoną o 1.
- IV 6.3 Napisz program, który wczyta dwie liczby i wypisze w oddzielnych wierszach pierwszą, drugą oraz ich różnicę.

### Poziom 2

- IV 6.4 Napisz program, który wczyta dwie liczby i wypisze pierwszą powiększoną o 3, literę  $x$  oraz w nowej linii drugą liczbę.
- IV 6.5 Napisz program, który wczyta dwie liczby i wypisze drugą powiększoną o 2 oraz w nowej linii dwukrotnie pierwszą pomniejszoną o 2.
- IV 6.6 Napisz program, który wczyta dwie liczby i wypisze napis  $7b$ . Następnie przejdzie do nowej linii i w oddzielnych wierszach wypisze drugą wczytaną liczbę pomniejszoną o 2 oraz pierwszą zwiększoną o 7.

### Poziom 3

- IV 6.7 Napisz program, który wczyta trzy liczby i wypisze w różnych wierszach trzecią zmniejszoną o 1, drugą zwiększoną o 2 i pierwszą zwiększoną o 3.
- IV 6.8 Napisz program, który wczyta trzy liczby. Następnie w oddzielnych wierszach wypisze sumę drugiej i trzeciej oraz różnicę pierwszej i drugiej.
- IV 6.9 Napisz program, który wczyta trzy liczby. Następnie w oddzielnych wierszach wypisze różnicę trzeciej i pierwszej oraz sumę obliczonej różnicy powiększoną o drugą.

### **Zadania dla mistrza**

- IV 6.10\* Napisz program, który wczyta trzy liczby, zwiększy każdą z nich o 3, a następnie wypisze powiększone liczby w osobnych wierszach.
- IV 6.11\* Pan bibliotekarz bardzo lubi czytać książki. Tego dnia przeczytał ich aż trzy. Książka, którą planuje przeczytać jutro, ma tyle stron co suma stron pierwszej i trzeciej książki pomniejszona o ilość stron drugiej książki. Napisz program, który wczyta, ile stron miała każda z przeczytanych dzisiaj przez Pana bibliotekarza książek i wypisze na konsoli, ile stron będzie miała następna książka.

## InstaKod 7

### Suma i różnica 1

#### Suma i różnica w pudełku pomocniczym. Przypisanie wartości do pudełka.



Komputer Bajtek ma do dyspozycji 4 pudełka. Czasem ma wykonać kilka operacji arytmetycznych i potrzebuje wielokrotnego dostępu do wczytanych przez użytkownika liczb, co oznacza, że nie powinien ich zmieniać. Co wtedy? Może wówczas wykonywać operacje arytmetyczne w innych pudełkach niż w tych, które zawierają wczytane liczby. Takie pudełka będziemy nazywać pomocniczymi.

**Pudełko pomocnicze** może służyć do liczenia sumy lub różnicy liczb znajdujących się w dwóch innych pudełkach. Korzystając z pudełka pomocniczego, nie zmieniamy wartości pudełek, do których wczytaliśmy dane.

#### Wejście i wyjście

Wejście to dane podane do programu. Wyjście to dane wypisywane przez program. Poznaliśmy poniższe instrukcje wejścia i wyjścia.

**Wczytaj do** B jest wejściem danych do programu. Przy każdym uruchomieniu programu instrukcja poprosi użytkownika o podanie liczby. Wczytana liczba zostanie zapisana w pudełku B. Liczba, która wcześniej znajdowała się w B, zostanie usunięta.

**Wypisz pudełko** C wypisuje na konsoli liczbę znajdującą się w pudełku C. Wypisana wartość może być zmienna i zależy od tego, jaka liczba aktualnie znajduje się w pudełku C.

**Wypisz napis** 25 & 26 wypisuje na konsoli tekst z ramki. Służy programiście do wyświetlania komunikatu dla użytkownika programu. Wypisywany komunikat jest stały i nie zależy od wczytanych danych.

**Przejdź do nowej linii** sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.



## Operacje arytmetyczne

Operacje arytmetyczne polegają na wykonywaniu działań arytmetycznych na liczbach. Wśród tych działań znajdują się dodawanie i odejmowanie.

**Zwiększ** D o 5 oblicza sumę liczby znajdującej się w pudełku D oraz liczby 5 i zapisuje wynik w pudełku D.

**Zmniejsz** A o B oblicza różnicę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku B i zapisuje wynik w pudełku A.

## Przypisanie wartości do pudełka

Przypisanie wartości do pudełka to zapisanie w nim podanej liczby lub liczby zapisanej w podanym pudełku.

**Ustaw** A na 5 przypisze do pudełka A liczbę 5. W pudełku A będzie znajdować się liczba 5.

**Ustaw** A na B przypisze do pudełka A liczbę, która znajduje się w pudełku B. W pudełku A i B będą znajdować się te same liczby.

## Przykład 1.

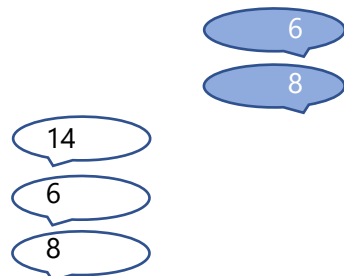
Napisz program, który wczyta dwie liczby do pudełek A i D, obliczy ich sumę w pudełku pomocniczym C i wypisze wynik. Następnie przejdzie do nowej linii i w oddzielnych wierszach wypisze wczytane liczby.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

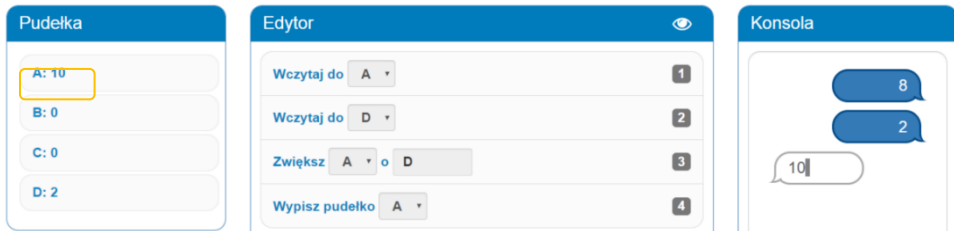
Wejście i wyjście

Dla podanych na wejściu liczb 6 i 8 program powinien na wyjściu wypisać 14 (ponieważ  $6 + 8 = 14$ ) oraz 6 i 8.

Konsola



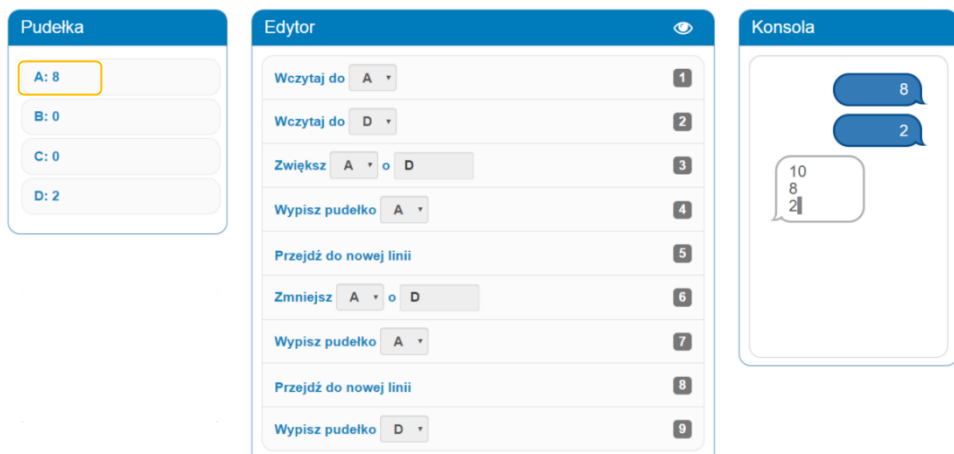
Rozwiązanie zadania będziemy budować stopniowo. Na początku policzymy sumę liczb z pudełek A i B tak, jak to robiliśmy w poprzednim rozdziale. Wczytajmy liczby do pudełek A i D i obliczmy sumę A i D w pudełku A. Ilustruje to poniższy program. Po uruchomieniu wczytajmy liczby 8 i 2. Pierwsza trafi do pudełka A, a druga do pudełka D. Po wykonaniu operacji **Zwiększ** A o D do pudełka A trafi suma  $8 + 2 = 10$ .



Zwróć uwagę, że w ten sposób tracimy informację o liczbie, która została wczytana do pudełka A. Ta informacja jest nam potrzebna, ponieważ w dalszej części zadania mamy wypisać wczytane liczby. Musimy zastosować inny sposób.

## Sposób 1.

Możemy odtworzyć liczbę wczytaną do A, wykonując odwrotną operację. W tym przypadku będzie to **Zmniejsz** A o D. Program wykona działanie  $10 - 2 = 8$  i zapisze wynik w pudełku A.



## Sposób 2

Innym sposobem jest obliczenie sumy  $A + D$  w jednym z dwóch pozostałych pudełek, np. w pudełku C. Takie pudełko będziemy od teraz nazywać pudełkiem pomocniczym.

Pudełko A: 8	0 0 0 0 1 0 0 0	Pierwsza wczytana liczba
Pudełko B: 0	0 0 0 0 0 0 0 0	
Pudełko C: 10	0 0 0 0 1 0 1 0	Pudełko pomocnicze Suma $A + D$ , czyli 10
Pudełko D: 2	0 0 0 0 0 0 1 0	Druga wczytana liczba

W jaki sposób policzyć sumę  $A$  i  $D$  w pudełku pomocniczym C?

Najpierw przepisujemy liczbę z pudełka A do C za pomocą instrukcji `Ustaw C na A`. Następnie zwiększamy nowo zapisaną liczbę w C o liczbę z pudełka B za pomocą `Zwiększ C o D`. W ten sposób obliczyliśmy sumę  $A + D$  w pudełku pomocniczym C, pozostawiając niezmienione liczby w pudełkach A i D.

Nasz program wygląda jak poniżej. Uruchommy go, wczytując liczby  $A = 8$  i  $D = 2$ . Przepisujemy liczbę z pudełka A do pudełka C

The image shows a screenshot of a programming environment with three panels: 'Pudełka' (Registers), 'Edytor' (Editor), and 'Konsola' (Console).

- Pudełka:** A: 8, B: 0, C: 8, D: 2. A blue arrow points from the 'A: 8' register to the 'C: 8' register.
- Edytor:** A list of instructions:
  - Wczytaj do A
  - Wczytaj do D
  - Ustaw C na A
  - Zwiększ C o D (highlighted with a yellow box)
  - Wypisz pudełko C
  - Przejdź do nowej linii
  - Wypisz pudełko A
  - Przejdź do nowej linii
  - Wypisz pudełko D
- Konsola:** Displays two output messages: 8 and 2.

Następnie zwiększamy C, w którym jest teraz liczba z pudełka A o liczbę z pudełka D. Obliczymy sumę  $A + D$  w pudełku pomocniczym C. Na końcu wypisujemy jeszcze wczytane liczby.

The screenshot shows a programming environment with three main panels:

- Pudełka (Boxes):** A: 8, B: 0, C: 10, D: 2.
- Edytor (Editor):** A list of 9 steps:
  - Wczytaj do A
  - Wczytaj do D
  - Ustaw C na A
  - Zwiększ C o D
  - Wypisz pudełko C
  - Przejdź do nowej linii
  - Wypisz pudełko A
  - Przejdź do nowej linii
  - Wypisz pudełko D
- Konsola (Console):** Shows the output of the program: 8, 2, and a list containing 10, 8, and 2.

Buttons at the bottom include: 'Zrób krok' (Make step), 'Edytuj' (Edit), 'Przetestuj' (Test), and 'Powrót do lekcji' (Return to lesson).

## Przykład 2.

Napisz program, który wczyta dwie liczby do pudełek A i B, obliczy ich sumę w pudełku pomocniczym C i wypisze wyliczoną sumę. Następnie obliczy ich różnicę w pudełku pomocniczym C i w nowym wierszu wypisze wyliczoną różnicę.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

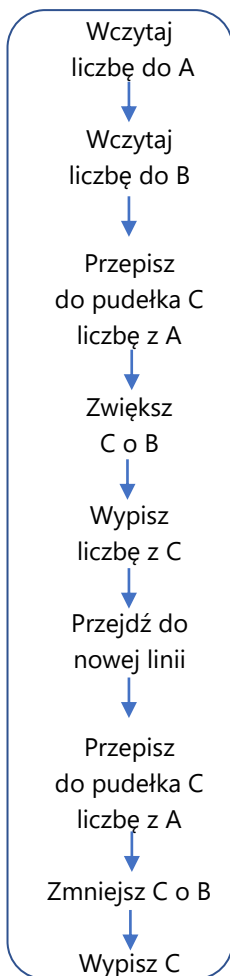
Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 4 i 2 program powinien wypisać na wyjściu 6 (ponieważ $4 + 2 = 6$ ) oraz 2 (ponieważ $4 - 2 = 2$ ).	<p>The console shows the input values 4 and 2, followed by the calculated sum 6 and the calculated difference 2 on separate lines.</p>

Mamy obliczyć sumę i różnicę dwóch wczytanych liczb, zachowując wczytane liczby. Zaplanujmy, gdzie zapiszemy liczby i gdzie wykonamy obliczenia.

Pudełko A: 4	0 0 0 0 0 1 0 0	Pierwsza wczytana liczba
Pudełko B: 2	0 0 0 0 0 0 1 0	Druga wczytana liczba
Pudełko C: 6	0 0 0 0 0 1 1 0	Pudełko pomocnicze Suma A + B, czyli 6
Pudełko C: 2	0 0 0 0 0 0 1 0	Pudełko pomocnicze Różnica A - B, czyli 2

Pudełko C wykorzystujemy dwukrotnie. Po obliczeniu sumy wypiszemy ją i nie będzie już nam potrzebna, dlatego można skorzystać z pudełka C drugi raz, by obliczyć różnicę.

Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Mamy wczytać dwie liczby. Zadanie precyzuje pudełka, do których je wczytujemy, dlatego wybieramy pudełka A i B.

Wczytaj do A

Wczytaj do B

W tym zadaniu mamy do obliczenia zarówno sumę, jak i różnicę. Aby obliczyć sumę i różnicę, użyjemy dwukrotnie pudełka pomocniczego C. Obliczamy sumę w pudełku pomocniczym C. Przepisujemy do pudełka C liczbę z A za pomocą instrukcji „Ustaw”. Zwiększamy liczbę w C o B. Suma A + B zostanie zapisana w C. Wypisujemy sumę na konsolę. Przechodzimy do nowej linii.

Ustaw C na A

Zwiększ C o B

Wypisz pudełko C

Przejdź do nowej linii

Drugi raz korzystamy z pudełka C, tym razem do obliczenia różnicy. Ustawiamy C na A, a następnie zmniejszamy C o B. Pozostało nam jeszcze wypisanie wyniku na konsolę.

Ustaw C na A

Zmniejsz C o B

Wypisz pudełko C

## Zadania. InstaKod Rozdział 7. Suma i różnica 1.

### Poziom 1

- IV 7.1 Napisz program, który wczyta dwie liczby do pudełek A i B, obliczy ich różnicę w pudełku pomocniczym i wypisze wynik. Następnie przejdzie do nowej linii i w oddzielnych wierszach wypisze wczytane liczby.
- IV 7.2 Napisz program, który wczyta dwie liczby do pudełek A i D, obliczy ich sumę w pudełku pomocniczym i wypisze wynik. Następnie przejdzie do nowej linii i w oddzielnych wierszach wypisze wczytane liczby.
- IV 7.3 Napisz program, który wczyta dwie liczby do pudełek A i B, obliczy ich różnicę w pudełku pomocniczym, a następnie w oddzielnych wierszach wypisze liczbę 3 oraz wyliczoną różnicę.

### Poziom 2

- IV 7.4 Napisz program, który wczyta dwie liczby do pudełek A i B oraz obliczy ich sumę i różnicę w pudełkach pomocniczych. Następnie w oddzielnych wierszach wypisze najpierw różnicę, a potem sumę.
- IV 7.5 Napisz program, który wczyta dwie liczby do pudełek A i C, obliczy ich sumę w pudełku pomocniczym i wypisze literę x oraz wyliczoną sumę. Następnie przejdzie do nowej linii i w oddzielnych wierszach wypisze wczytane liczby.
- IV 7.6 Napisz program, który wczyta dwie liczby do pudełek A i B i obliczy ich różnicę w pudełku pomocniczym. Następnie w oddzielnych wierszach wypisze różnicę zwiększoną o 5 oraz wczytane liczby.

### Poziom 3

- IV 7.7 Napisz program, który wczyta dwie liczby do pudełek B i C oraz obliczy ich sumę w pudełku pomocniczym. Następnie w oddzielnych wierszach wypisze dwukrotność obliczonej sumy oraz wczytane liczby.
- IV 7.8 Napisz program, który wczyta dwie liczby do pudełek A i B oraz obliczy ich różnicę w pudełku pomocniczym. Następnie w oddzielnych wierszach wypisze liczbę 3, obliczoną różnicę zwiększoną o 5 oraz wczytane liczby.
- IV 7.9 Napisz program, który wczyta dwie liczby do pudełek D i A oraz obliczy ich różnicę w pudełku pomocniczym. Następnie w oddzielnych wierszach wypisze obliczoną różnicę zwiększoną o 3, liczbę 11 oraz wczytane liczby pomniejszone o 1.

### Zadania dla mistrza

- IV 7.10\* Napisz program, który wczyta cztery liczby do pudełek C, B, D i A. Następnie w oddzielnych wierszach wypisze różnicę drugiej i trzeciej, pierwszej i czwartej oraz sumę wczytanych liczb.
- IV 7.11\* Śmieci pozostawione przez ludzi na plaży przeszkadzały wszystkim jej mieszkańcom. Kraby postanowiły zaprowadzić porządek, dzieląc się na dwie drużyny. Pod koniec dnia okazało się, że pierwsza drużyna zebrała więcej śmieci niż druga. Napisz program, który wczyta do pudełek A i B, ile śmieci zebrała pierwsza drużyna krabów, a ile druga. Następnie w osobnych wierszach wypisze, o ile więcej śmieci zebrała pierwsza drużyna oraz liczbę śmieci zebranych przez obie drużyny.

## InstaKod 8

### Suma i różnica 2

#### Suma i różnica w pudełku pomocniczym. Krok dalej.

Pudełko pomocnicze to miejsce w pamięci komputera służące do wykonywania obliczeń. W Assembly mamy dostępne 4 miejsca, do których możemy zapisać wczytane liczby lub wyniki obliczeń. Są to pudełka: A, B, C i D.

Jeśli dane wejściowe zapiszemy w pudełkach A i B, to do obliczeń możemy wykorzystać pudełka C i D.

Przy rozwiązywaniu zadań musimy najpierw zaplanować, do których pudełek wczytamy liczby i które pudełka wykorzystamy do obliczeń. Możemy to zrobić za pomocą tabelki z pudełkami.

Załóżmy, że chcemy wczytać dwie liczby i obliczyć ich sumę.

Zaplanujmy, gdzie zapiszemy liczby i gdzie wykonamy działania.

	Sposób I Możemy wczytać liczby do A i B, a obliczenia wykonać w pudełku C.	Sposób II Możemy też wczytać liczby do pudełek B i D, a obliczenia wykonać w pudełku A.
Pudełko A	Pierwsza wczytana liczba	Suma wczytanych liczb
Pudełko B	Druga wczytana liczba	Pierwsza wczytana liczba
Pudełko C	Suma wczytanych liczb	
Pudełko D		Druga wczytana liczba

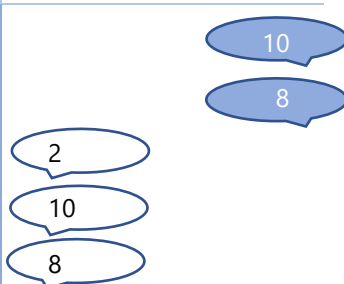
Zastanów się, jakie są inne możliwości zapisania liczb i wyboru pudełka pomocniczego.



## Przykład 1.

Napisz program, który wczyta dwie liczby, a następnie w oddzielnych wierszach wypisze ich różnicę oraz wczytane liczby.

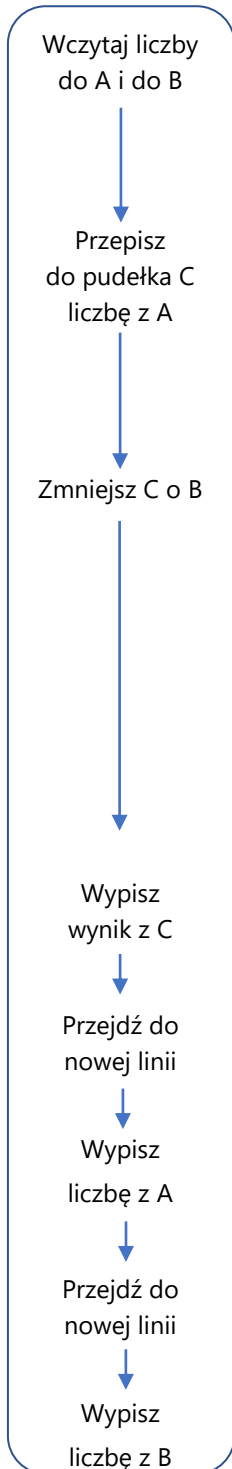
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 10 i 8 program powinien wypisać na wyjściu 2 (ponieważ $10 - 8 = 2$ ) oraz 10 i 8.	

Mamy obliczyć różnicę dwóch wczytanych liczb, nie zmieniając wartości pudełek, do których wczytamy liczby. Zaplanujmy, gdzie zapiszemy liczby i gdzie wykonamy obliczenia.

Pudełko A: 10	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	1	0	1	0	Pierwsza wczytana liczba
0	0	0	0	1	0	1	0			
Pudełko B: 8	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	1	0	0	0	Druga wczytana liczba
0	0	0	0	1	0	0	0			
Pudełko C: 2	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	0	1	0	Pudełko pomocnicze Różnica A - B, czyli 2
0	0	0	0	0	0	1	0			

## Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Zaplanowaliśmy wczytanie liczb do A i B. Używamy instrukcji „Wczytaj” oraz wybranych pudełek.

Wczytaj do A

Wczytaj do B

Mając do dyspozycji dane wejściowe, możemy przejść do wykonywania operacji arytmetycznych.

Ustaw C na A

Zmniejsz C o B

Mamy do obliczenia różnicę, którą zdecydowaliśmy, że policzymy w pudełku pomocniczym C. Dzięki temu zachowamy wczytane liczby w pudełkach A i B.

Przepisujemy pierwszą wczytaną liczbę z pudełka A do pudełka C instrukcją `Ustaw C na A`.

Zmniejszamy zapisaną w C pierwszą wczytaną liczbę o wartość drugiej wczytanej liczby, która jest w pudełku B.

Możemy teraz przejść do wypisania wyniku oraz wczytanych liczb. Wypisujemy pudełko C, gdzie znajduje się obliczona różnica.

Wypisz pudełko C

Przejdź do nowej linii

Wypisz pudełko A

Przejdź do nowej linii

Przechodzimy do nowej linii.

Następnie wypisujemy pudełko A, w którym jest pierwsza wczytana liczba i przechodzimy do nowej linii.

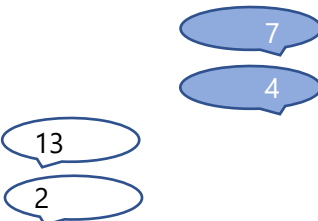
Wypisz pudełko B

Na koniec wypisujemy pudełko B, w którym jest druga wczytana liczba.

## Przykład 2.

Napisz program, który wczyta dwie liczby oraz wypisze w osobnych wierszach ich sumę zwiększoną o 2 i różnicę pomniejszoną o 1.

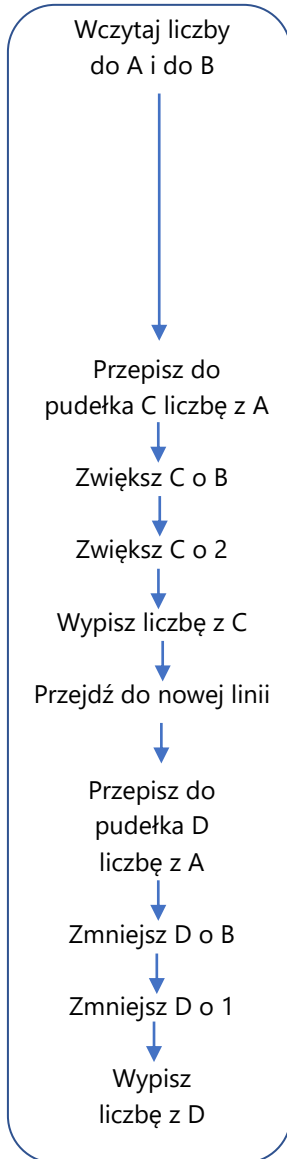
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 7 i 4 program powinien w oddzielnych liniach wypisać na wyjściu 13 (ponieważ $7 + 4 + 2 = 13$ ) oraz 2 (ponieważ $7 - 4 - 1 = 2$ ).	

Mamy obliczyć sumę wczytanych liczb zwiększoną o 2 i różnicę wczytanych liczb pomniejszoną o 1, nie zmieniając wartości pudełek, do których wczytamy liczby. Zaplanujmy, gdzie zapiszemy liczby i gdzie wykonamy obliczenia.

Pudełko A: 7	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	0	0	1	1	1	Pierwsza wczytana liczba
0	0	0	0	0	1	1	1			
Pudełko B: 4	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	1	0	0	Druga wczytana liczba
0	0	0	0	0	1	0	0			
Pudełko C: 13	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>	0	0	0	0	1	1	0	1	Pudełko pomocnicze Suma $A + B + 2$
0	0	0	0	1	1	0	1			
Pudełko D: 2	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	0	1	0	Pudełko pomocnicze Różnica $A - B - 1$
0	0	0	0	0	0	1	0			

## Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Mamy wczytać dwie liczby. Zadanie nie precyzuje pudełek, do których je wczytujemy. W przykładzie wybrane zostały pudełka A i B. Używamy instrukcji „Wczytaj” oraz odpowiednich pudełek.

```
Wczytaj do A
Wczytaj do B
```

Użyjemy pudełka pomocniczego C do obliczenia  $A + B + 2$ .

Przepisujemy do C wartość z A, korzystając z instrukcji

```
Ustaw C na A
```

Zwiększamy C o B, a następnie zwiększamy C o 2.

Możemy wypisać wynik na konsolę. Następnie przechodzimy do nowej linii.

```
Ustaw C na A
Zwiększ C o B
Zwiększ C o 2
Wypisz pudełko C
Przejdź do nowej linii
```

Użyjemy pudełka pomocniczego D do obliczenia różnicy wczytanych liczb pomniejszonej o 1.

Przepisujemy do D wartość z A, korzystając z instrukcji

```
Ustaw D na A
```

Zmniejszamy D o B. Następnie zmniejszamy D o 1. Możemy wypisać wynik na konsolę.

```
Ustaw D na A
Zmniejsz D o B
Zmniejsz D o 1
Wypisz pudełko D
```

## Zadania InstaKod 8. Suma i różnica 2.

### Poziom 1

- IV 8.1 Napisz program, który wczyta dwie liczby i w oddzielnych wierszach wypisze ich sumę oraz liczby w kolejności wczytania.
- IV 8.2 Napisz program, który wczyta dwie liczby i w oddzielnych wierszach wypisze ich różnicę powiększoną o 1 oraz drugą wczytaną liczbę.
- IV 8.3 Napisz program, który wczyta dwie liczby, a następnie w oddzielnych wierszach wypisze literę  $x$ , różnicę wczytanych liczb powiększoną o 12 oraz wczytane liczby.

### Poziom 2

- IV 8.4 Napisz program, który wczyta dwie liczby. Następnie wypisze w osobnych wierszach ich sumę oraz dwukrotnie pierwszą wczytaną liczbę.
- IV 8.5 Napisz program, który wczyta dwie liczby. Następnie wypisze w osobnych wierszach ich sumę, drugą wczytaną liczbę oraz sumę wczytanych liczb powiększoną o 3.
- IV 8.6 Napisz program, który wczyta dwie liczby i wypisze w osobnych wierszach sumę zwiększoną o 5 oraz drugą liczbę pomniejszoną o 1.

### Poziom 3

- IV 8.7 Napisz program, który wczyta dwie liczby, a następnie w oddzielnych wierszach wypisze ich różnicę zwiększoną o 2 oraz wczytane liczby pomniejszone o 1.
- IV 8.8 Napisz program, który wczyta dwie liczby, a następnie w oddzielnych wierszach wypisze napis  $abc$ , sumę wczytanych liczb pomniejszoną o 1 oraz ich różnicę.
- IV 8.9 Napisz program, który wczyta dwie liczby i wypisze w osobnych wierszach ich sumę, różnicę oraz drugą wczytaną liczbę.

## Zadania dla mistrza

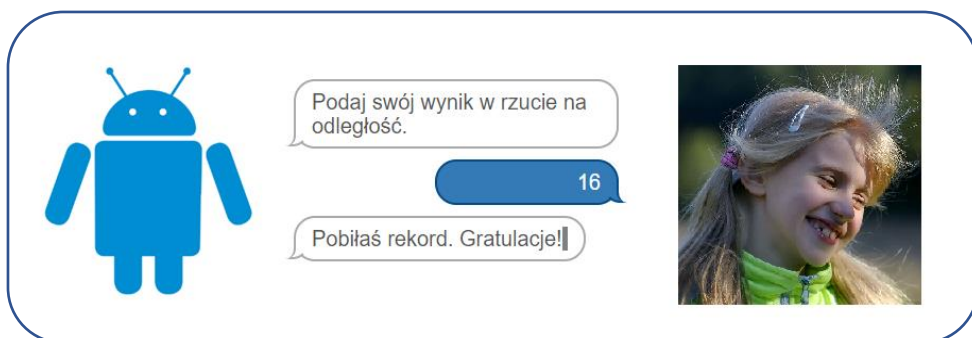
- IV 8.10\* Napisz program, który wczyta cztery liczby następnie w oddzielnych wierszach wypisze ich sumę oraz sumę ich dopełnień do 10.
- IV 8.11\* Na urodziny imperatora przybyło wielu znamienitych gości. Z gwiazdy IC 4592 z Mgławicy Niebieski Koński Łeb i NGC 6334 z Mgławicy Kocia Łapa przybyły dwie grupy dostojników. Siedmiu spośród dostojników z IC 4592 przybyło na urodziny wraz ze współmałżonkami. Napisz program, który wczyta, ilu dostojników z IC 4592 przybyło na urodziny imperatora, a ilu z NGC 6334. Następnie w osobnych wierszach wypisze, ile osób liczyła delegacja IC 4592 (dostojnicy i współmałżonkowie), ile łącznie dostojników przybyło na urodziny króla oraz ilu dostojników przybyło z gwiazdy NGC 6334.

## InstaKod 9

### Instrukcja warunkowa 1

Pierwsze starcie. Jeśli warunek jest spełniony, to podejmij działanie, w przeciwnym razie zakończ program.

W niektórych zadaniach będziemy musieli uzależnić wykonanie instrukcji od spełnienia przez dane wejściowe pewnego warunku. Część instrukcji zostanie wykonana tylko wtedy, gdy dane wejściowe spełnią określony przez nas warunek.




Komputer Bajtek ma poinformować, czy twoja koleżanka pobiła rekord w rzucaniu piłką tenisową na odległość (aktualny rekord szkoły wynosił 12 m). Aby to było możliwe, musisz napisać program sterujący Bajtkiem, który po wczytaniu długości rzutu twojej koleżanki wypisze „Pobiłaś rekord”, jeśli to prawda.

Do napisania takiego programu będziesz potrzebować instrukcji warunkowej! Jak zmodyfikujesz program, jeśli któryś z twoich kolegów pobije rekord? Z tym zadaniem możesz zmierzyć się w dziale „Zadania dla mistrza”.

## Przykład 1.

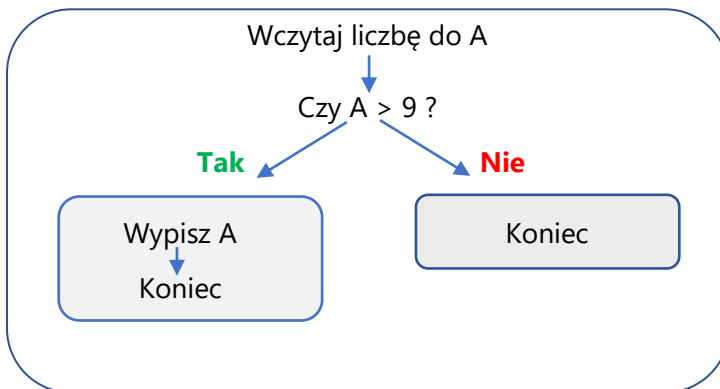
Napisz program, który wypisze wczytaną liczbę, jeśli jest większa od 9.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 20 program powinien wypisać na wyjściu 20.	

Jak sprawdzić, która liczba jest większa? Trzeba zrobić porównanie wczytanej liczby i 9. Zapiszmy to w diagramie.

Na pytanie „Czy  $A > 9$ ?” możemy odpowiedzieć **tak** lub **nie**. Diagram rozwiązania zadań z warunkami tworzymy przy użyciu strzałek. Każda ze strzałek reprezentuje inną odpowiedź na postawione w problemie pytanie. Taki diagram nazywamy **drzewem**, a jego strzałki **gałęziami**.



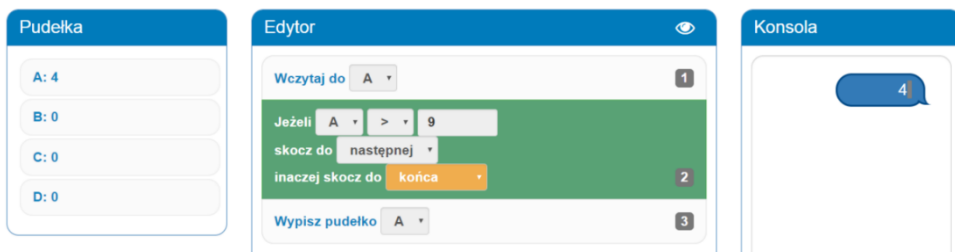
Do sprawdzania wszelkich porównań używamy w Assembly instrukcji „Jeżeli”. Dla powyższego przykładu porównaniem jest  $A > 9$ . Jeśli **warunek jest spełniony**, przechodzimy (skaczemy) do następnej linii kodu, gdzie wypisujemy wartość pudełka A i kończymy program. Jeśli **warunek nie jest spełniony**, kończymy program (skaczemy do końca).



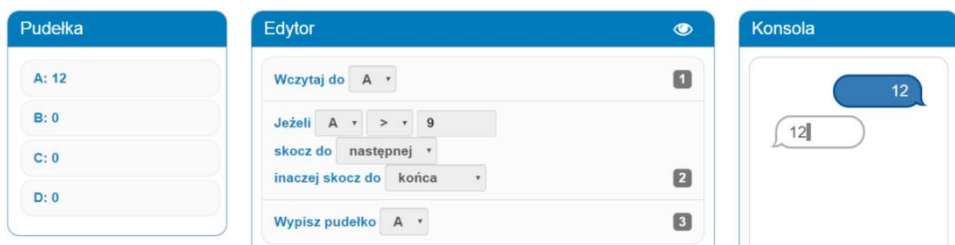
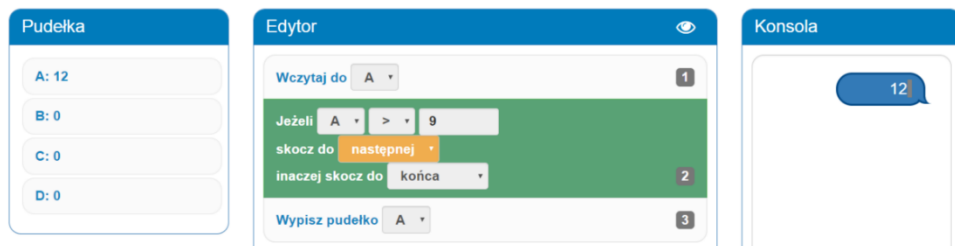
Rozwiązanie przeniesione z diagramu do edytora Assembly będzie wyglądało następująco:

```
Wczytaj do A
Jeżeli A > 9 skocz do następnej
inaczej skocz do końca
Wypisz pudełko A
```

Uruchommy nasz program. W momencie wykonania instrukcji „Jeżeli” komputer sprawdza warunek i podświetla na pomarańczowo jedno z dwóch pól. Dla wczytanej liczby 4 warunek nie jest spełniony ( $4 \leq 9$ ). Na pomarańczowo podświetlone będzie pole „koniec”. Program ominie instrukcję `Wypisz pudełko A` i zakończy działanie.



Dla wczytanej liczby 12 warunek jest spełniony ( $12 > 9$ ). Na pomarańczowo podświetlone będzie pole „następna”. Program nie ominie instrukcji `Wypisz pudełko A` i zakończy działanie dopiero po jej wykonaniu.



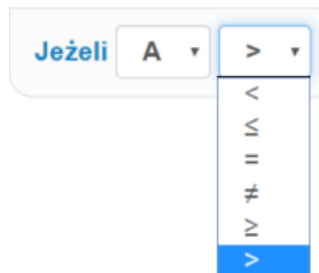
Instrukcja „Jeżeli” składa się z trzech osobnych części:

1	2	3
Warunek	Informacja, do której linii kodu program ma skoczyć, jeśli warunek <b>jest</b> spełniony.	Informacja, do której linii kodu program ma skoczyć, jeśli warunek <b>nie jest</b> spełniony.

Warunek w Assembly to porównanie jednego z czterech pudełek z innym pudełkiem bądź liczbą.

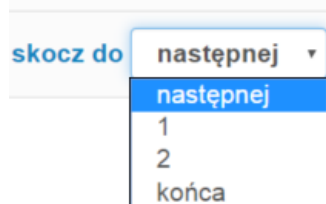
Dostępne są następujące porównania:

- < mniejsze niż
- ≤ mniejsze lub równe
- = równe
- ≠ nie równe, czyli różne
- > większe
- ≥ większe lub równe



Skok w instrukcji warunkowej to informacja o numerze wiersza, do którego przejdzie program. Możemy wybrać:

- następną linię kodu,
- numer wiersza,
- koniec.



Zadaliśmy pytanie: „Czy A jest większe od 9?”

Jeśli odpowiedź brzmi **tak**, to przechodzimy do następnej linii kodu.

Jeśli odpowiedź brzmi **nie**, to kończymy działanie programu.

Kiedy warunek  $A > 9$  nie jest spełniony?

Nie jest spełniony wtedy, gdy prawdziwa jest relacja przeciwna, czyli  $A \leq 9$  (czytamy: A jest mniejsze od 9 lub A jest równe 9).

Na przykład, jeśli w pudełku A jest liczba 2, to relacja  $A \leq 9$  jest prawdziwa, ponieważ  $2 \leq 9$ .

### Instrukcja warunkowa

Instrukcja warunkowa umożliwia uzależnienie wykonania fragmentu programu od spełnienia pewnego warunku.

**Jeżeli**  $A > 9$  **skocz do** następnej **inaczej skocz do** końca

Jeżeli warunek  $A > 9$  **jest** spełniony, to wykonana zostanie instrukcja z następnej linii kodu. Jeżeli warunek **nie jest** spełniony, (to znaczy, że  $A < 9$  lub  $A = 9$ ), program zostanie zakończony.

### Instrukcja „skocz do”

**Skocz do** 5 przechodzi do podanego numeru linii kodu, w tym przypadku do linii 5. Pozwala przeskoczyć blok instrukcji lub wrócić do bloku instrukcji już wykonanych.

**Skocz do** końca kończy program.

## Przykład 2.

Napisz program, który wczyta liczbę i sprawdzi, czy jest różna od 10. Jeśli nie, to wypisze wczytaną liczbę pomniejszoną o 1 oraz literę x.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.



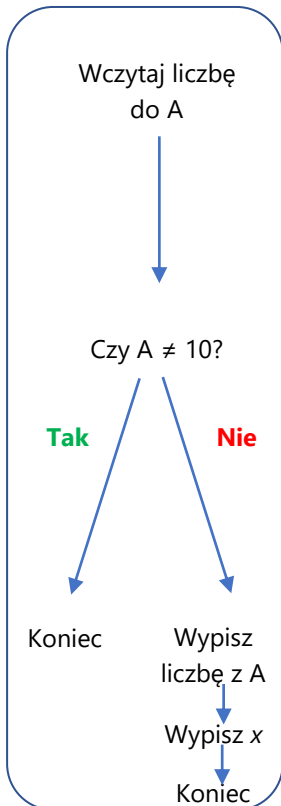
Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 10 program powinien sprawdzić, że warunek $10 \neq 10$ nie jest spełniony, i wypisać na wyjściu 9 (ponieważ $10 - 1 = 9$ ) oraz literę x.	 <pre>10 9x</pre>
Dla podanej na wejściu liczby 14 program powinien sprawdzić, że warunek $14 \neq 10$ jest spełniony i zakończyć działanie.	 <pre>14</pre>

Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Zadanie nie precyzuje pudełka, do którego mamy wczytać liczbę. Wybieramy je sami. Decydujemy się na pudełko A.

```
Wczytaj do A
```

Mając dane, możemy przejść do porównania. Mamy sprawdzić, czy wczytana liczba jest różna od 10. Zapisujemy warunek  $A \neq 10$ . Jeśli warunek jest spełniony, tzn. jeśli  $A \neq 10$ , kończymy nasz program. Jeśli jest inaczej, czyli gdy  $A = 10$ , skaczemy do następnej linii kodu i wykonujemy zapisaną tam instrukcję.

```
Jeżeli A ≠ 10
skocz do końca
inaczej skocz
do następnej
```

Mamy wypisać wczytaną liczbę pomniejszoną o 1 oraz literę x. Najpierw zmniejszamy liczbę w A o 1, a następnie ją wypisujemy. Na końcu wypisujemy literę x.

```
Zmniejsz A o 1
Wypisz pudełko A
Wypisz napis x
```

### Przykład 3.

Napisz program, który wczyta dwie liczby i sprawdzi, czy są równe. Jeśli tak, to wypisze ich różnicę.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

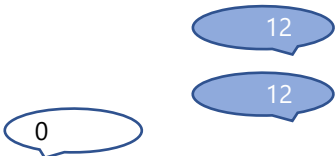
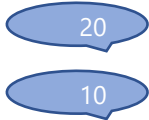
Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 12 i 12 program powinien sprawdzić, że warunek $12 = 12$ jest spełniony i wypisać na wyjściu różnicę $12 - 12$ , czyli 0.	
Dla podanych na wejściu liczb 20 i 10 program powinien sprawdzić, że warunek $20 = 10$ nie jest spełniony i zakończyć działanie.	

Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Mamy wczytać dwie liczby. Zadanie nie precyzuje pudełek, do których wczytujemy liczby. Wybieramy pudełka A i B.

```

Wczytaj do A
Wczytaj do B
  
```

Czy wczytane liczby są równe? Zapisujemy warunek  $A = B$ .

```

Jeżeli A = B skocz
do następnej
inaczej skocz
do końca
  
```

Jeśli  $A = B$ , skaczemy do następnej linii kodu i wykonujemy dalszą część programu. Jeśli  $A \neq B$ , kończymy program.

Pozostało obliczenie i wypisanie różnicy. Różnicę możemy obliczyć w pudełku A, zmniejszając A o B. Następnie wypisujemy wynik na konsolę.

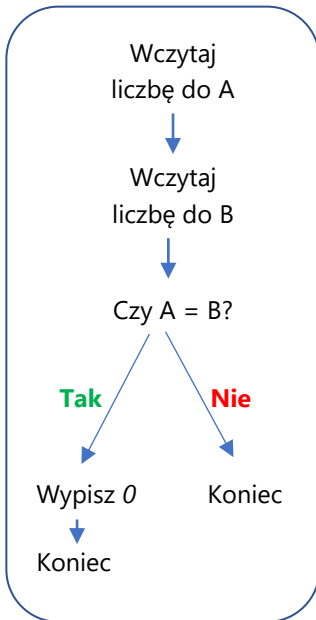
```

Zmniejsz A o B
Wypisz pudełko A
  
```

Rozwiązaliśmy zadanie poprawnie, ale istnieje inny, prostszy sposób jego wykonania.

Jeśli liczby są równe, to ich różnica zawsze będzie równa 0!

Wystarczy więc sprawdzić, czy liczby są równe. Jeśli tak, to należy wypisać 0, jeśli nie, to należy zakończyć program.



Wczytaj do A

Wczytaj do B

Jeżeli A = B skocz do następnej  
inaczej skocz do końca

Wypisz napis 0

W informatyce często spotyka się zadania, których rozwiązania można bardzo uprościć, jeśli zauważone zostaną pewne zależności, tak jak w tym przypadku. Jeśli zauważyłaś / zauważyłeś, że program albo wypisuje 0, albo kończy działanie, to gratulacje!

## Zadania InstaKod 9. Instrukcja warunkowa 1.

### Poziom 1

- IV 9.1 Napisz program, który wczyta liczbę i sprawdzi, czy jest mniejsza od 15. Jeśli tak, to wypisze ją na ekran.
- IV 9.2 Napisz program, który wczyta liczbę i sprawdzi, czy jest równa 7. Jeśli nie, to wypisze *Nie*.
- IV 9.3 Napisz program, który wczyta liczbę i sprawdzi, czy jest mniejsza lub równa 3. Jeśli tak, to wypisze ją na ekran.

### Poziom 2

- IV 9.4 Napisz program, który wczyta liczbę i sprawdzi, czy jest różna od 8. Jeśli tak, to wypisze ją zwiększoną o 5.
- IV 9.5 Napisz program, który wczyta dwie liczby i sprawdzi, czy druga jest większa od pierwszej. Jeśli tak, wypisze napis *Tak*.
- IV 9.6 Napisz program, który wczyta dwie liczby i sprawdzi, czy są równe. Jeśli tak, to wypisze drugą liczbę zwiększoną o 3.

### Poziom 3

- IV 9.7 Napisz program, który wczyta dwie liczby i sprawdzi, czy pierwsza jest większa od 7. Jeśli tak, to w oddzielnych wierszach wypisze pierwszą i drugą zwiększoną o 2.
- IV 9.8 Napisz program, który wczyta dwie liczby i sprawdzi, czy suma pierwszej i liczby 5 jest większa od drugiej wczytanej liczby. Jeśli tak, to w oddzielnych wierszach wypisze wczytane liczby w kolejności wczytania.
- IV 9.9 Napisz program, który wczyta dwie liczby i sprawdzi, czy różnica drugiej i liczby 3 jest mniejsza od pierwszej wczytanej liczby. Jeśli tak, to wypisze różnicę wczytanych liczb.

## Zadania dla mistrza

- IV 9.10\* Napisz program, który wczyta dwie liczby i jeśli ich suma jest większa od 12, to wypisze wczytane liczby w oddzielnych wierszach.
- IV 9.11\* Chcesz by komputer Bajtek pomógł ci w odpowiadaniu kolejnym uczniom czy pobili rekord szkoły w rzucie piłką tenisową na odległość. Napisz program, który wyświetli komunikat: *Jaki jest aktualny rekord szkoły?* oraz czyta od użytkownika liczbę. Następnie wyświetli komunikat: *Podaj swój wynik w rzucie piłką tenisową* oraz czyta od użytkownika liczbę. Jeśli rekord został pobity, program powinien wyświetlić komunikat: *Gratulacje! Udało ci się pobić rekord szkoły.*



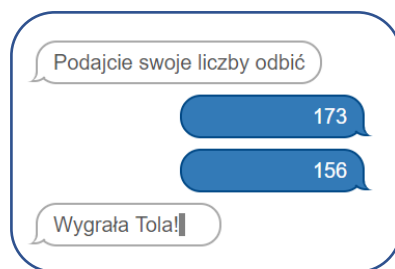
## InstaKod 10

### Instrukcja warunkowa 2

Jeśli warunek jest spełniony, to podejmij działanie I, w przeciwnym razie podejmij działanie II.

Przy użyciu instrukcji warunkowej możemy tworzyć bardziej rozbudowane programy. W tym rozdziale przyjrzymy się programom, które wykonują różne działania w zależności od tego, czy warunek jest spełniony, czy też nie.

Ty i twoja koleżanka Tola w ramach treningu piłki nożnej i poprawy techniki żonglujecie piłką. Komputer Bajtek ma być waszym sędzią i ogłosić zwycięzcę. Jaki warunek powinien sprawdzić Bajtek?



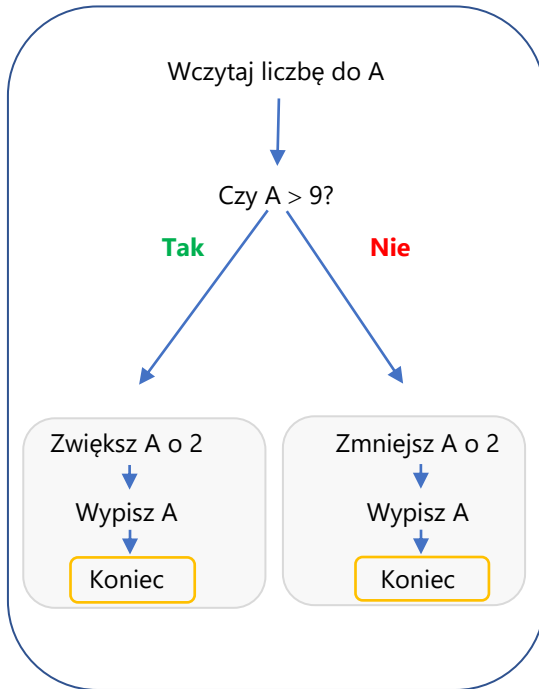
#### Przykład 1.

Napisz program, który wczyta liczbę oraz sprawdzi, czy jest ona większa od 9. Jeśli tak, to doda do niej 2 i wypisze wynik. W przeciwnym razie odejmie 2 i wypisze wynik.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 12 program powinien sprawdzić, że warunek $12 > 9$ jest spełniony i wypisać na wyjściu 14 (ponieważ $12 + 2 = 14$ ).	<p>12</p> <p>14</p>
Dla podanej na wejściu liczby 5 program powinien sprawdzić, że warunek $5 > 9$ nie jest spełniony i wypisać na wyjściu 3 (ponieważ $5 - 2 = 3$ ).	<p>5</p> <p>3</p>

## Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie od wczytania danych. W przykładzie wybrane zostało pudełko A.

Pytamy, czy wczytana liczba jest większa od 9. Symboliczny zapis wygląda następująco: Czy  $A > 9$ ? Mamy dwie możliwe odpowiedzi: **tak** lub **nie**.

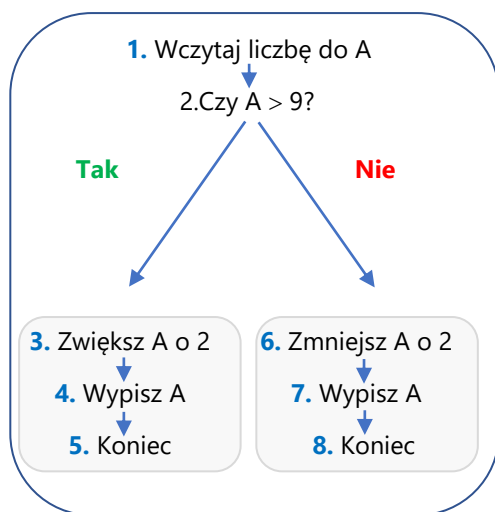
Jeśli odpowiedzią jest **tak**, liczymy sumę wczytanej liczby i liczby 2. Następnie wypisujemy wynik na konsolę.

Jeśli odpowiedzią jest **nie**, liczymy różnicę wczytanej liczby i liczby 2. Następnie wypisujemy wynik na konsolę.

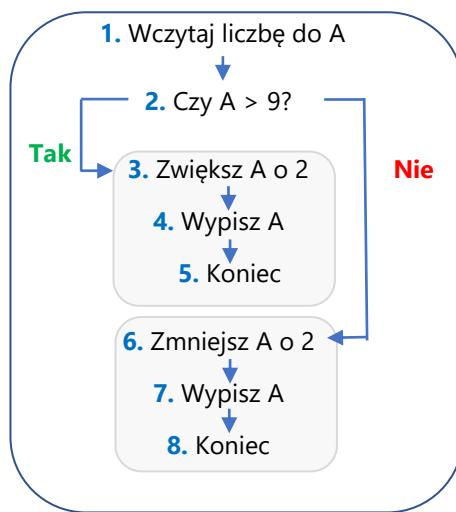
Zwróć uwagę, że oba bloki instrukcji zakończone zostały słowem **Koniec**. Jeśli nie oddzielimy końcem dwóch bloków instrukcji, to program wykona oba bloki instrukcji, jeden po drugim.

Kod programu pisany jest linia po linii. Przed implementacją diagram w formie drzewa musimy przetłumaczyć na [diagram liniowy](#).

W tym celu ponumerujemy kroki w diagramie drzewa.



Następnie przepisujemy do diagramu liniowego kroki w takiej kolejności, w jakiej zostały ponumerowane w diagramie drzewa.



Zwróć uwagę, że ostatni krok nie jest potrzebny na końcu programu.

Zaimplementujemy diagram liniowy w edytorze Assembly, a następnie uruchomimy nasz program. Zrobimy pierwszy krok i wczytajmy liczbę 15. Warunek  $15 > 9$  **jest** spełniony (program skoczy do następnej linii). Wykonany zostanie blok z linii 3-5. Liczba w pudełku A zostanie zwiększona o 2 i na konsolę zostanie wypisany wynik. Następnie program zakończy działanie.

The screenshot shows an assembly editor interface with three main panels:

- Pudełka (Registers):** A: 15, B: 0, C: 0, D: 0.
- Edytor (Code Editor):** A list of assembly instructions:
  - 1. Wczytaj do A
  - 2. Jeżeli A > 9, skocz do następnej, inaczej skocz do 6
  - 3. Zwiększ A o 2
  - 4. Wypisz pudełko A
  - 5. Skocz do końca
  - 6. Zmniejsz A o 2
  - 7. Wypisz pudełko A
- Konsola (Console):** Displays the number 15.

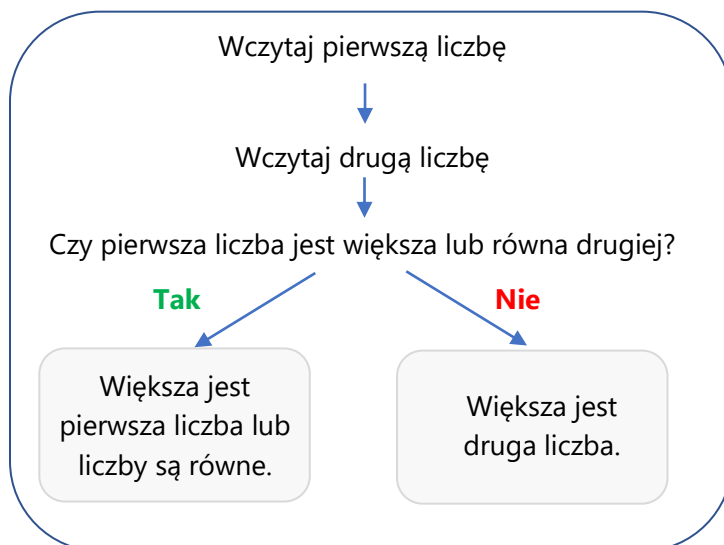
Dla wczytanej liczby 8, warunek  $8 < 9$  **nie jest** spełniony. Wykonany zostanie blok z linii 6-7. Liczba w pudełku A zostanie zmniejszona o 2 i na konsolę zostanie wypisany wynik. Następnie program zakończy działanie.

The screenshot shows a programming environment with three panels: 'Pudełka' (Variables), 'Edytor' (Editor), and 'Konsola' (Console).  
- 'Pudełka': A: 8, B: 0, C: 0, D: 0.  
- 'Edytor':  
 1. Wczytaj do A  
 2. Jeżeli A > 9 skocz do następnej, inaczej skocz do 6  
 3. Zwiększ A o 2  
 4. Wypisz pudełko A  
 5. Skocz do końca  
 6. Zmniejsz A o 2 (highlighted in yellow)  
 7. Wypisz pudełko A  
- 'Konsola': 8

## Przykład 2.

Napisz program, który wczyta dwie liczby i w oddzielnych wierszach wypisze najpierw większą, a potem mniejszą. Jeśli wczytane liczby są równe, program w oddzielnych wierszach wypisze wczytane liczby.

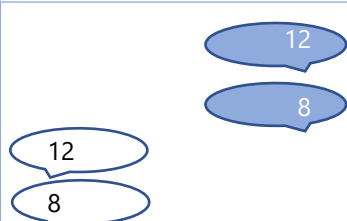
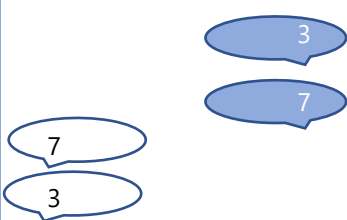
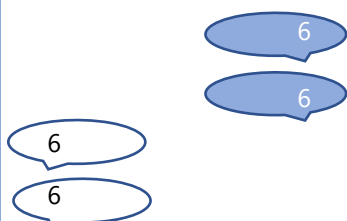
Pytanie, które zadajemy to: „Czy pierwsza liczba jest większa lub równa drugiej?” Jeśli warunek jest spełniony, to większa jest pierwsza liczba lub liczby są równe.



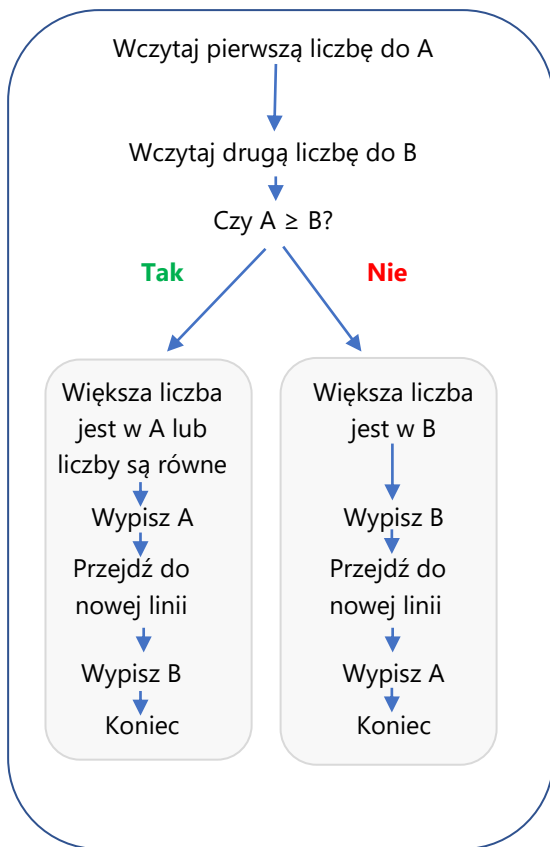
Uwaga: W zadaniach z porównaniem dwóch liczb mamy trzy przypadki:

1. Pierwsza liczba jest większa od drugiej, np. jeśli wczytamy liczby 9 i 5.
2. Pierwsza liczba jest mniejsza od drugiej, np. jeśli wczytamy liczby 7 i 11.
3. Dwukrotnie wczytano tę samą liczbę, np. jeśli dwukrotnie wczytamy liczbę 5.

Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 12 i 8 program powinien wypisać na wyjściu najpierw 12, a następnie 8.	 <p>The terminal shows two lines of input: 12 and 8. The output consists of two lines: 12 and 8. The input and output values are highlighted with blue speech bubbles.</p>
Dla podanych na wejściu liczb 3 i 7 program powinien wypisać na wyjściu najpierw 7, a następnie 3.	 <p>The terminal shows two lines of input: 7 and 3. The output consists of two lines: 7 and 3. The input and output values are highlighted with blue speech bubbles.</p>
Dla podanych na wejściu liczb 6 i 6 program powinien wypisać na wyjściu w oddzielnych wierszach dwukrotnie liczbę 6.	 <p>The terminal shows two lines of input: 6 and 6. The output consists of two lines: 6 and 6. The input and output values are highlighted with blue speech bubbles.</p>

## Diagram rozwiązania i jego implementacja.



Wczytujemy dwie liczby.  
Wybieramy pudełka A i B.

Zadajemy pytanie:  
„Czy pierwsza liczba jest większa lub równa drugiej?”.  
Zapisujemy to symbolicznie:  
Czy  $A \geq B$ ? Mamy dwie możliwe odpowiedzi - **tak** lub **nie**.

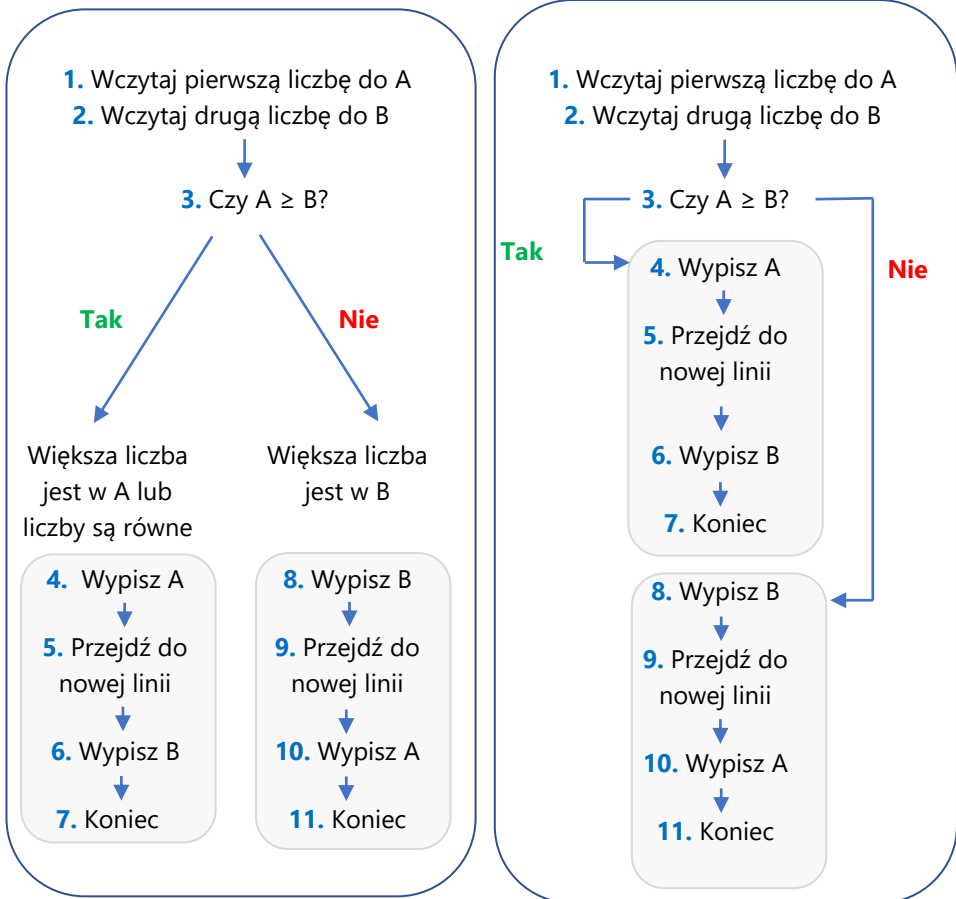
Wypisujemy najpierw większą liczbę, a potem w nowym wierszu mniejszą i kończymy program.

Aby zaimplementować rozwiązanie, musimy je przepisać na kod linii po linii.

W tym celu tłumaczymy diagram drzewa na diagram liniowy.

Numerujemy kroki w diagramie drzewa.

Następnie przepisujemy do diagramu liniowego kroki w takiej kolejności, w jakiej zostały ponumerowane w diagramie drzewa.



Zwróć uwagę, że ostatni krok nie jest potrzebny na końcu programu.

Teraz z łatwością przepisujemy diagram liniowy na kod:

Wczytaj do	A	1	x		
Wczytaj do	B	2	x		
Jeżeli	A	≥	B	3	x
skocz do	następnej	inaczej skocz do	8		
Wypisz pudełko	A	4	x		
Przejdź do nowej linii		5	x		
Wypisz pudełko	B	6	x		
Skocz do	końca	7	x		
Wypisz pudełko	B	8	x		
Przejdź do nowej linii		9	x		
Wypisz pudełko	A	10	x		



## Zadania InstaKod 10. Instrukcja warunkowa 2.

### Poziom 1

- IV 10.1 Napisz program, który wczyta liczbę i sprawdzi, czy jest równa 3. Jeśli tak, wypisze tę liczbę. Jeśli nie, to wypisze napis *Nie*.
- IV 10.2 Napisz program, który wczyta liczbę i sprawdzi, czy jest mniejsza od 15. Jeśli tak, wypisze ją. Jeśli nie, wypisze ją zwiększoną o 2.
- IV 10.3 Napisz program, który wczyta liczbę i sprawdzi, czy jest większa lub równa 5. Jeśli tak, wypisze *Tak*. W przeciwnym razie wypisze *Nie*.

### Poziom 2

- IV 10.4 Napisz program, który wczyta dwie liczby i wypisze mniejszą.
- IV 10.5 Napisz program, który wczyta dwie liczby i sprawdzi, czy pierwsza jest większa od drugiej. Jeśli tak, to wypisze w osobnych wierszach pierwszą i drugą. W przeciwnym razie wypisze w osobnych wierszach drugą i pierwszą.
- IV 10.6 Napisz program, który wczyta dwie liczby i sprawdzi, czy są równe. Jeśli tak, wypisze *Tak*, w przeciwnym razie wypisze ich sumę.

### Poziom 3

- IV 10.7 Napisz program, który wczyta dwie liczby i sprawdzi czy ich różnica jest mniejsza od 5. Jeśli tak, wypisze liczbę 3, w przeciwnym razie wypisze drugą wczytaną liczbę.
- IV 10.8 Napisz program, który wczyta dwie liczby i sprawdzi czy ich suma jest większa od 10. Jeśli tak, wypisze *Tak*, w przeciwnym razie wypisze różnicę wczytanych liczb.
- IV 10.9 Napisz program, który wczyta dwie liczby i wypisze w osobnych wierszach mniejszą, a potem większą.

## Zadania dla mistrza:

IV 10.10\* Napisz program, który wczyta trzy liczby i w oddzielnych wierszach wypisze większą spośród pierwszej i drugiej oraz dwukrotność trzeciej.

IV 10.11\* Próżna księżniczka Tafah liczyła szmaragdy w swoich naszyjnikach. Chce wybrać ten, który ma ich najwięcej. Napisz program, który wczyta, ile szmaragdów jest w pierwszym naszyjniku, ile w drugim i w oddzielnych wierszach dwukrotnie wypisze większą liczbę szmaragdów.

## Pojęcia informatyczne oraz instrukcje języka Assembly

### Zmienna

Pudełko jest miejscem w pamięci komputera, w którym przechowujemy liczbę. Pudełko ma swoją nazwę i rozmiar (liczbę bitów). W Assembly pudełka nazywają się: A, B, C i D. Każde ma 16 bitów oraz przypisaną wartość początkową 0.

### Wejście programu

Wejście to dane podane do programu przez użytkownika.

**Wczytaj do B** jest **wejściem** danych liczbowych do programu. Przy każdym uruchomieniu programu instrukcja poprosi użytkownika o podanie liczby. Wczytana liczba zostanie zapisana w pudełku B. Liczba, która wcześniej znajdowała się w B, zostanie usunięta.

### Wyjście programu, stałe i zmienne

Wyjście to dane wypisane przez program w celu komunikacji z użytkownikiem.

Wyjście stałe jest zawsze takie samo, niezależne od wejścia.

Wyjście zmienne jest różne, zależne od podanych danych wejściowych.

W Assembly mamy trzy instrukcje wyjścia. Są to:

**Wypisz pudełko C** wypisuje na konsoli liczbę znajdującą się w pudełku C. Wypisana wartość może być zmienna i zależy od tego, jaka liczba aktualnie znajduje się w pudełku C.

**Wypisz napis** 2345 wypisuje na konsoli tekst z ramki. Służy programiście do wyświetlania komunikatu dla użytkownika programu. Wypisywany komunikat jest stały i nie zależy od wczytanych danych.

**Przejdź do nowej linii** sprawia, że kolejny zapis na konsoli pojawi się w nowym wierszu.

### Operacje arytmetyczne

Operacje arytmetyczne polegają na wykonywaniu działań arytmetycznych na liczbach. Wśród tych działań znajdują się dodawanie i odejmowanie.

W Assembly wszystkie operacje arytmetyczne wykonywane są na liczbach całkowitych. Do budowania wyrażeń arytmetycznych używamy dwóch instrukcji - „Zwiększ” oraz „Zmniejsz”.

Dodawanie

**Zwiększ** D o 5 oblicza sumę liczby znajdującej się w pudełku D oraz liczby 5 i zapisuje wynik w pudełku D.

Odejmowanie

**Zmniejsz** A o B oblicza różnicę liczby znajdującej się w pudełku A oraz liczby znajdującej się w pudełku B i zapisuje wynik w pudełku A.

### Przypisanie wartości do pudełka

Przypisanie wartości do pudełka to zapisanie w nim podanej liczby lub liczby zapisanej w podanym pudełku.

Instrukcja **Ustaw** służy do przypisania (wstawienia) do wybranego pudełka konkretnej liczby lub wartości innego pudełka.

**Ustaw** A na 5 przypisze do pudełka A liczbę 5. W pudełku A będzie znajdować się liczba 5.

**Ustaw** A na B przypisze do pudełka A liczbę, która znajduje się w pudełku B. W pudełku A będzie znajdować się ta sama liczba co w pudełku B.

### Instrukcja warunkowa

Instrukcja warunkowa umożliwia uzależnienie wykonania fragmentu programu od spełnienia pewnego warunku.

Jeżeli A > 9 skocz do następnej inaczej skocz do końca

Jeżeli warunek A > 9 **jest** spełniony, to wykonana zostanie instrukcja z następnej linii kodu. Jeżeli warunek **nie jest** spełniony (to znaczy, że A < 9 lub A = 9), program zostanie zakończony.

Jeżeli B ≠ 5 skocz do 7 inaczej skocz do następnej

Jeżeli warunek B ≠ 5 **jest** spełniony, to wykonana zostanie instrukcja z 7 linii kodu. Jeżeli warunek **nie jest** spełniony (to znaczy, że B = 5), wykonana zostanie instrukcja z następnej linii kodu.

### Instrukcja skocz do

**Skocz do** 5 przechodzi do podanego numeru linii kodu, w tym przypadku do linii 5. Pozwala przeskoczyć blok instrukcji lub wrócić do bloku instrukcji już wykonanych.

**Skocz do** końca kończy program.

# TECHNOLOGIE INFORMACYJNE

Ucząc się, komunikując ze znajomymi czy też rozwijając swoje hobby, korzystasz z wielu różnych aplikacji komputerowych.

Technologie informacyjne to obszerny zbiór zagadnień informatyki zajmujący się przetwarzaniem informacji. Nasze działania ograniczą się do pracy w 4 typach aplikacji, które będą użyteczne nie tylko w realizowaniu różnorodnych zadań z języka polskiego, historii, przyrody czy innych przedmiotów szkolnych, ale również w rozwijaniu pasji i kreatywności.

Są to:

- program graficzny,
- edytor tekstu,
- program do prezentacji,
- arkusz kalkulacyjny.

Twoja nauka wyżej wymienionych aplikacji oparta będzie na podejściu projektowym. Celem zdobywania kolejnych umiejętności w posługiwaniu się wymienionymi aplikacjami jest wykorzystanie ich do zrealizowania projektów międzyprzedmiotowych zaproponowanych na końcu każdego rozdziału lub zdefiniowanych przez twojego nauczyciela. Projektem możemy nazwać zbiór powiązanych ze sobą czynności, prowadzących do realizacji z góry zamierzonego celu.

Każdy z projektów możesz wykonać jako łączną pracę na zajęciach informatyki i innego przedmiotu szkolnego, wykorzystując wybraną aplikację danego typu.

Każdy projekt może zostać oceniony podwójnie – pod względem technicznym na zajęciach informatyki oraz pod względem merytorycznym na przedmiocie, którego dotyczy.

Aplikacji, w których możesz wykonać projekt jest wiele, niemniej w ramach jednego typu aplikacji mają one podobne funkcje i w podobny sposób się z nich korzysta. W każdym rozdziale omówione zostały wybrane funkcje jednego lub dwóch wybranych przez nas programów danego typu.

Specjalnie dla Ciebie, do każdego rozdziału przygotowaliśmy również ćwiczenia wspomagające korzystanie ze wspomnianych aplikacji. Osiągnięta biegłość w posługiwaniu się nimi, pozwoli Ci na skuteczną realizację zadanych projektów.

Powodzenia!

# 1

## Edytor graficzny

### Malowanie kreską, krzywą i kształtami.

### Rysowanie linii i krzywych.

Na tej lekcji nauczysz się:

1. uruchamiać program graficzny;
2. tworzyć własne kształty, korzystając z narzędzia linii i krzywej;
3. zapisywać utworzone rysunki na dysku.

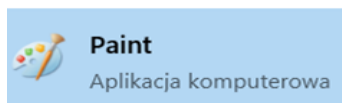
### Uruchamianie programu

Proces uruchamiania programu będzie nieznacznie różnił się w zależności od wersji systemu operacyjnego zainstalowanego na twoim komputerze.

Przykładowo, dla systemu Windows 10 możliwe jest znalezienie dowolnego zainstalowanego programu w Menu Start:

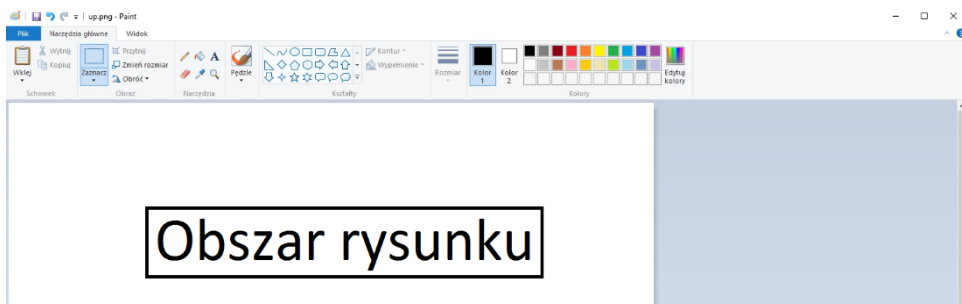


1. Najedź kursorem na symbol znajdujący się w lewym dolnym rogu ekranu.
2. W polu wyszukiwania wpisz *Paint*.
3. Kliknij w ikonę programu.



### Pusty projekt

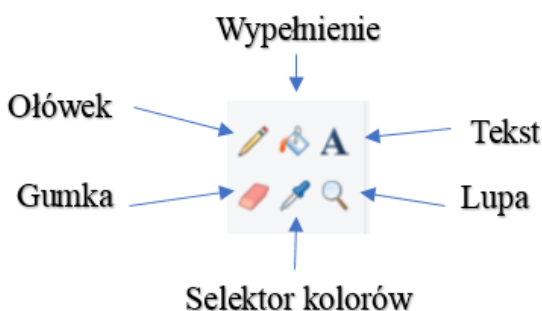
Po uruchomieniu aplikacji pojawi się okno programu, składające się z dwóch części: wstążki (znajdują się na niej trzy karty – Plik, Narzędzia główne oraz Widok) i obszaru rysunku.



1. Wymiary rysunku możesz zmienić poprzez przeciągnięcie jednego z kwadracików znajdujących się na brzegu obszaru rysunku.
2. Aby zmniejszyć wymiary obrazka, należy przytrzymać wciśnięty lewy przycisk myszy i przeciągnąć do wewnątrz.
3. Aby zwiększyć wymiary obrazka, należy przytrzymać wciśnięty lewy przycisk myszy i przeciągnąć na zewnątrz.

## Podstawowe narzędzia

W górnej części aplikacji znajdziesz kartę *Narzędzia główne*, w skład której wchodzi: Schowek, Obraz, Narzędzia, Kształty, Kolory, Pędzle i Rozmiary. Narzędzia umożliwią ci tworzenie i edycje rysunków na ekranie komputera. Oto podstawowe narzędzia:

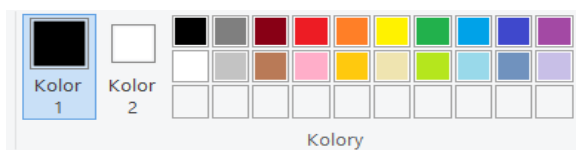


### Ołówek

Narzędzie to służy do rysowania. Gdy wypróbujesz jego działanie, szybko przekonasz się, że rysuje się nim dokładnie tak jak zwykłym ołówkiem na kartce papieru.

Przydatne opcje w pracy z ołówkiem to zmiana koloru i grubość linii ołówka.

Kolor ołówka zmienisz, korzystając z palety kolorów znajdującej się w górnej części aplikacji, w karcie *Narzędzia główne*.



**Kolor 1** to kolor, którym będziesz rysować ołówkiem i malować pędzlem.

**Kolor 2** to kolor tła, który służy również do wypełniania kształtów, o czym wkrótce się przekonasz.



Kliknij w ikonę **Kolor 1**, a następnie zaznacz wybrany przez siebie kolor, by użyć go do pisania ołówkiem.

Aby zmienić grubość linii ołówka, kliknij na ikonkę **Rozmiar** znajdującą się w górnej części aplikacji, w karcie Narzędzia główne.

Dostępne są cztery grubości linii ustawione w kolejności od najcieńszej do najgrubszej.



### Wypełnienie kolorem

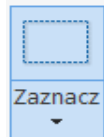
Wypełnienie kolorem służy do wypełniania całych rysunków lub pojedynczych kształtów wybranym kolorem. Kolor wypełnienia wybiera się w ten sam sposób co kolor ołówka.



### Gumka

Gumka służy do wymazania części obrazu. Za pomocą gumki możesz zetrzeć wszystko, co narysowałeś/narysowałaś. Wielkość gumki zmienia się w podobny sposób co grubość ołówka.

### Narzędzie wycinania



Narzędzie wycinania umożliwia zaznaczenie dowolnego obszaru rysunku, a następnie usunięcie, przesunięcie lub skopiowanie zaznaczonego obszaru.

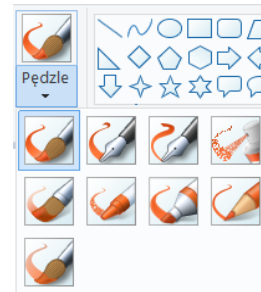


Po wybraniu opcji *Zaznacz* wciśnij i przytrzymaj lewy przycisk myszy. Ruszając myszą, utworzysz prostokąt wokół obszaru, który chcesz zaznaczyć.

Czynność	Skutek
Wciśnięcie klawisza Delete (czyt. dilit)	Usunięcie obszaru zaznaczenia
Przeciągnięcie myszą do innej części rysunku	Przesunięcie zaznaczenia do wskazanego obszaru
Wciśnięcie klawiszy Ctrl + C	Skopiowanie obszaru zaznaczenia

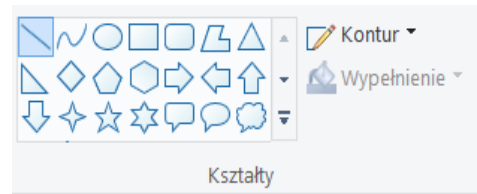
## Pędzle

Pędzel, podobnie jak ołówek, służy do rysowania. Pędzle umożliwiają rysowanie za pomocą różnych kresek. Przy pracy z pędzlem dostępne są opcje zmiany grubości oraz koloru, które działają tak samo jak dla ołówka.








## Kształty

W górnej części aplikacji znajdziesz grupę narzędzi Kształty. Korzystanie z gotowych figur geometrycznych może się okazać przydatne przy tworzeniu własnych rysunków.




Aby skorzystać z gotowych kształtów, wybierz figurę, którą chcesz narysować, a następnie rodzaj konturu oraz rodzaj wypełnienia. Wciśnij lewy przycisk myszy i przeciągając po obszarze rysunku, narysuj wybraną figurę.

Do kształtów należą:

- linia 
- krzywa 
- okrąg 
- prostokąt 
- trójkąt 

### Linia

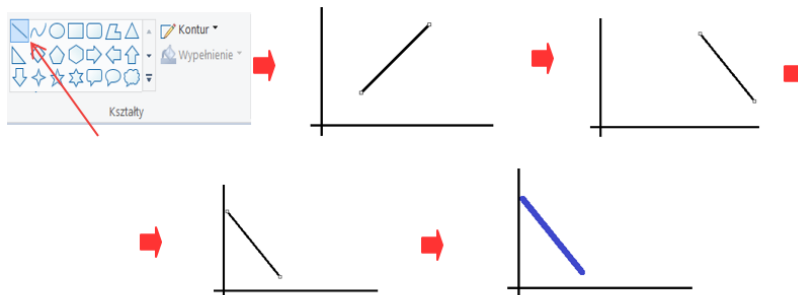
Aby narysować odcinek za pomocą narzędzia *Linia*, należy wybrać w górnym panelu programu symbol , a następnie nacisnąć lewy przycisk myszy i przeciągnąć po obszarze rysunku.

W jaki sposób zmienić długość, nachylenie i położenie narysowanej linii?


1. Najedź myszą na kwadrat znajdujący się na końcu narysowanego odcinka, a następnie wciśnij lewy przycisk myszy. Poruszając myszą, zmienisz nachylenie i długość linii.

2. Najedź myszą na dowolny punkt wewnątrz odcinka, a następnie wciśnij lewy przycisk myszy. Poruszając myszą, zmienisz położenie linii.
3. Kolor i grubość linii zmienia się w ten sam sposób, co kolor i grubość ołówka.

Jeśli po narysowaniu linii klikniesz lewym przyciskiem myszy na dowolny punkt poza nią, nie będziesz już miał możliwości jej edytowania.



## Krzywa

Aby narysować krzywą za pomocą narzędzia *Krzywa*, należy wybrać w górnym panelu programu symbol , a następnie nacisnąć lewy przycisk myszy i przeciągnąć po obszarze rysunku.

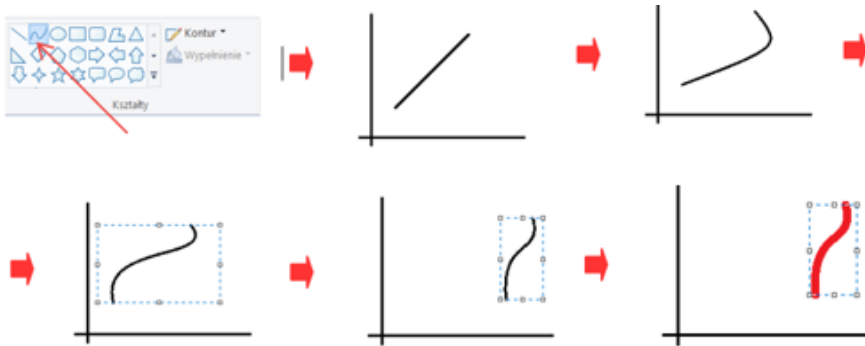
Po narysowaniu linii za pomocą narzędzia *Krzywa* nie ma możliwości zmiany jej długości. Istnieje możliwość zmiany kształtu narysowanej linii.

W celu zmiany kształtu należy najechać myszą i kliknąć na dowolny punkt na linii, a następnie przesunąć kursor myszy.

Po zakończeniu edycji krzywej pojawi się ramka. Opcje dostępne na tym etapie to:

- przeniesienie stworzonej krzywej poprzez przytrzymanie kursorem w dowolnym punkcie wewnątrz ramki i poruszanie myszą;
- rozciągnięcie lub zwinięcie ramki wraz z krzywą przy użyciu kwadratów w rogach i na bokach ramki;
- zmiana koloru i grubości krzywej, która odbywa się podobnie jak zmiana koloru i grubości ołówka.


Po skończeniu edycji nie ma możliwości wprowadzenia dalszych zmian.



## ↩ Cofnij

Każdą czynność w programie graficznym można cofnąć, wciskając kombinację klawiszy Ctrl+Z. Opcji *Cofnij* używaj zawsze wtedy, gdy popełniłeś/popełniłaś błąd i chcesz wrócić do poprzedniej wersji rysunku. Co ciekawe, opcji tej można użyć nawet 50 razy z rzędu!

## Zapisz projekt

Jeśli nie chcesz utracić swojej pracy, musisz zapisać gotowy projekt. W górnym rogu ekranu znajdziesz symbol dyskietki . Po naciśnięciu na niego pojawi się okno *Zapisz jako*. Możesz teraz utworzyć nowy folder lub wybrać już istniejący, w którym chcesz zapisać projekt. Musisz jeszcze nadać unikalną nazwę dla projektu i będziesz mógł kliknąć na pole *Zapisz*.

Otwórz pusty projekt w edytorze grafiki Paint.

## Zadanie 1.

Korzystając z podstawowych narzędzi programu Paint (ołówki, wypełnienie, gumka, pędzel, linia, krzywa), wykonaj rysunek świątecznej choinki. Za przykład może posłużyć ci rysunek znajdujący się obok.



Zapisz projekt w swoim folderze. Plik nazwij „Świąteczna\_choinka\_imię\_nazwisko\_klasa”.

Rys. 2

## Zadanie 2.

Za pomocą Linii i Krzywej, narysuj:

- literę A w kolorze czerwonym za pomocą linii,
- literę C w kolorze niebieskim za pomocą krzywej,
- lód w kolorze fioletowym za pomocą linii i krzywej,
- dzwon w kolorze zielonym za pomocą linii i krzywej.

Przykładowe rysunki:



Zapisz projekt w swoim folderze. Plik nazwij „Kształty\_imię\_nazwisko\_klasa”.

# Rysowanie kształtów. Powielanie wzorów. Pola tekstowe.

Na tej lekcji nauczysz się:

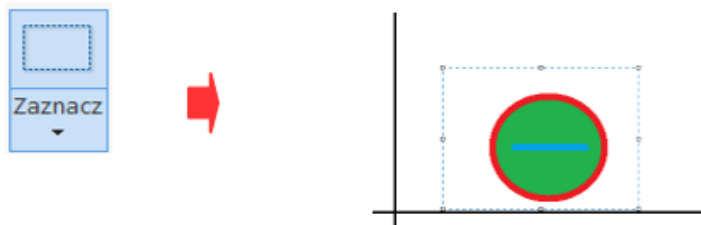
1. kopiować, przenosić i wycinać fragmenty rysunków;
2. korzystać z gotowych kształtów;
3. dodawać pole tekstowe i z niego korzystać;
4. przenosić rysunek na inny komputer.

## Narzędzie wycinania c.d.

Narzędzie wycinania poznaliśmy już w poprzednim rozdziale. W tym rozdziale dowiemy się nieco więcej o jego możliwościach.

Kliknij w ikonę *Zaznacz*, a następnie zaznacz kursorem obszar, który chcesz przenieść, wyciąć lub skopiować.

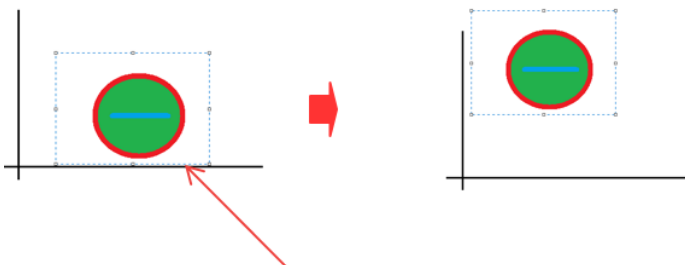
Pojawi się ramka odpowiadająca zaznaczonemu obszarowi:



Zaznaczenie obszaru umożliwia szereg czynności.

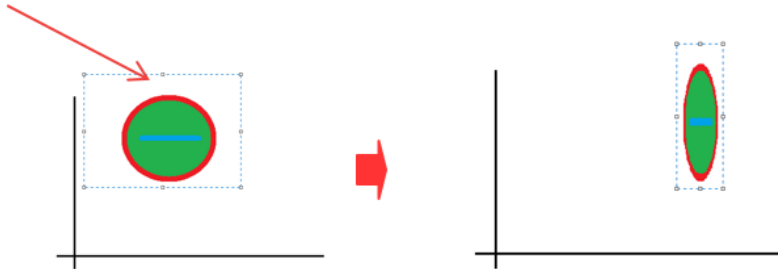
### Przeniesienie zaznaczonego fragmentu

Wystarczy złapać kursorem za dowolny punkt wewnątrz ramki, a następnie przesunąć myszą zaznaczony obszar.



## Zmiana kształtu i wielkości zaznaczonego fragmentu

Wystarczy złapać kursorem za jeden z kwadratów na krawędzi zaznaczonego obszaru, a następnie za pomocą myszy rozciągać lub zwęzić ten fragment.



## Skopiowanie lub usunięcie danego fragmentu

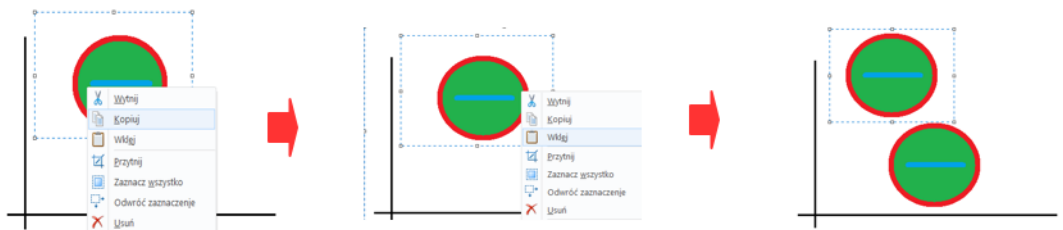
Aby usunąć wybrany obszar, należy wcisnąć klawisz Delete.

Aby go skopiować należy:

- zaznaczyć obszar i kliknąć w dowolnym jego miejscu prawym przyciskiem myszy;
- z dostępnych opcji wybrać *Kopiuj*;
- kliknąć prawym przyciskiem myszy w dowolnym miejscu projektu i wybrać *Wklej*.

Obu operacjom odpowiadają również skróty klawiszowe:

Ctrl+C – Kopiuj i Ctrl+V – Wklej.



## Kształty geometryczne

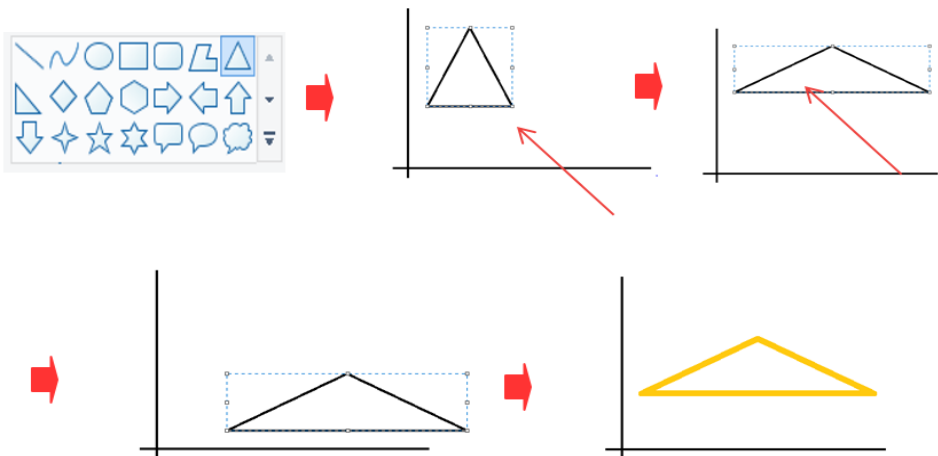
W poprzednim rozdziale była mowa o *Linii* i *Krzywej*. Inne, często wykorzystywane kształty, to: okrąg, prostokąt czy trójkąt.

W górnej części aplikacji znajdziesz kartę *Kształty*. Wybierz figurę, którą chcesz narysować, a następnie rodzaj konturu oraz rodzaj wypełnienia. Wciskając lewy przycisk myszy i przeciągając po obszarze rysunku, narysujesz figurę, która została zaznaczona.

Dostępne opcje to:

- zmiana kształtu figury poprzez przytrzymanie kursorem jednego z kwadratów na krawędziach obramowania i przeciągnięcie myszą;
- przeniesienie figury poprzez przytrzymanie kursorem dowolnego punktu wewnątrz obramowania i poruszenie myszą;
- zmiana koloru i grubości krzywej, która odbywa się w taki sam sposób jak zmiana koloru i grubości ołówka.

Gdy uznasz, że figura jest już gotowa, kliknij lewym przyciskiem myszy na dowolny punkt poza obszarem obramowania.

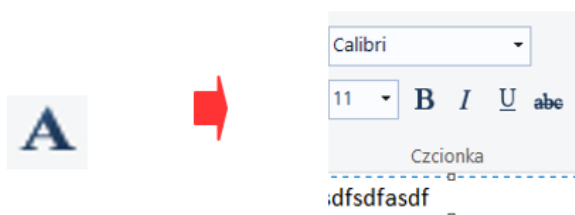




## A Pole tekstowe

Narzędzie to służy do dodawania tekstu do rysunku. Aby z niego skorzystać, wybierz narzędzie *Pola Tekstowego* **A** z karty *Narzędzia*. Następnie naciśnij lewym przyciskiem myszy na dowolny punkt rysunku, w którym chcesz stworzyć pole tekstowe.

Pojawi się ramka, do której możesz wpisać swój tekst. Zostanie również otwarta karta edycji tekstu, gdzie dostępne są opcje wyboru formatowania, rodzaju i wielkości czcionki.



Kształt i położenie ramki możemy zmieniać w taki sam sposób, jak w przypadku rysowania figur geometrycznych. Zmiana koloru tekstu następuje w taki sam sposób, w jaki zmienia się kolor ołówka.

## Przenośny dysk USB

Bardzo przydatnym narzędziem w pracy na komputerze jest przenośny dysk USB. Umożliwia on przenoszenie plików między komputerami. Jeśli chciałbyś/chciałabyś zanieść do domu rysunek zapisany w pliku na komputerze szkolnym, możesz go zapisać na dysku USB.

Aby to zrobić, wykonaj następujące kroki:

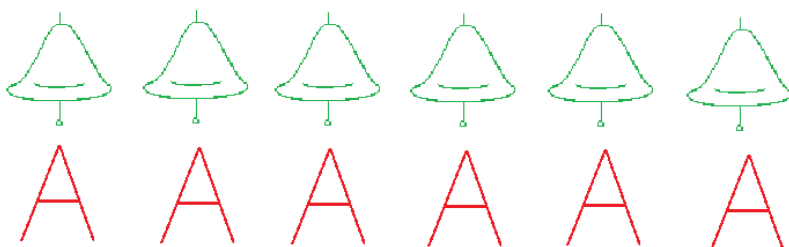
1. Zanim wsuniesz dysk USB do szkolnego komputera, poproś nauczyciela o uzyskanie zgody. Nauczyciel powinien sprawdzić twój dysk USB pod kątem szkodliwego oprogramowania.
2. Wejść w *Mój komputer* i znajdź dysk USB.
3. Znajdź plik, który chcesz skopiować i przenieś go na USB, korzystając z opcji *Kopiuj* i *Wklej*.
4. Upewnij się, że skopiowany plik znajduje się na dysku USB.
5. Bezpiecznie wysuń dysk przenośny.

## Zadanie 1.

Otwórz stworzony na ostatnich zajęciach plik „Kształty\_imię\_nazwisko\_klasa”.

Korzystając z narzędzia *Wycinania*, przenieś stworzone przez siebie figury na górę kartki tak, żeby mieć miejsce na kolejne rysunki.

Następnie, korzystając z tego samego narzędzia, stwórz pod istniejącymi obrazkami dwa rzędy identycznych, narysowanych przez siebie na poprzednich zajęciach figur, kopiując te istniejące i przenosząc je. Pamiętaj, że narzędzie *Wycinania* umożliwia zmianę wielkości stworzonego kształtu. Oto przykładowy rysunek:

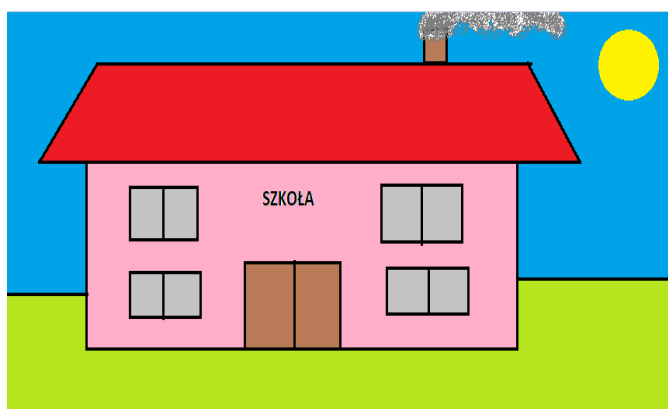


Rys. 3

## Zadanie 2.

Korzystając z figur geometrycznych, pola tekstowego oraz poznanych wcześniej narzędzi, stwórz rysunek przedstawiający twoją szkołę.

Za przykład może ci posłużyć poniższy rysunek:



Rys. 4

Zapisz stworzony rysunek w swoim folderze. Plik nazwij „Moja\_Szkoła\_Imię\_Nazwisko\_Klasa”.

Pora, by wykorzystać w praktyce poznane umiejętności. Poniżej znajdziesz propozycje projektów łączących informatykę i zajęcia plastyki. Zrealizuj jeden z nich lub inny zaproponowany przez twojego nauczyciela.

## Projekt międzyprzedmiotowy

### Edytor graficzny

### Informatyka / plastyka „Góry”

Korzystając z poznanych narzędzi i funkcji programu graficznego, stwórz rysunek pt.: „Góry”. Inspiracje możesz zaczerpnąć z własnych górskich wycieczek, znalezionych zdjęć lub filmów.



Rys. 5

Przy tworzeniu rysunku wykorzystaj poznane na lekcji narzędzia:

- linię i krzywą,
- gotowe kształty geometryczne,
- wypełnienie kolorem,
- ołówek i przynajmniej jeden z pędzli,
- narzędzie wycinania,
- pole tekstowe do podpisania się na pracy w prawym dolnym rogu.

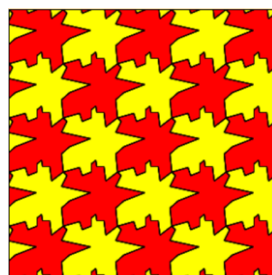
Gotową pracę przynieś na przenośnym dysku USB na lekcje informatyki oraz plastyki.

## Projekt międzyprzedmiotowy

### Edytor graficzny

#### Informatyka / plastyka „Mozaika”

Druga propozycja projektu to wykorzystanie edytora graficznego do stworzenia mozaiki. Inspiracją dla Twojej pracy może być kalejdoskop lub nazwana na cześć holenderskiego malarza i grafika Mauritsa Eschera, mozaika Eschera, składająca się z powtarzających się, regularnych wzorów.



Rys. 6

Stwórz własną mozaikę, korzystając z edytora graficznego.

Przy jej tworzeniu wykorzystaj poznane na lekcji narzędzia:

- linię i krzywą,
- gotowe kształty geometryczne,
- wypełnienie kolorem,
- narzędzie wycinania,
- pole tekstowe do podpisania się na pracy w prawym dolnym rogu.

Gotową pracę przynieś na przenośnym dysku USB na lekcje informatyki oraz plastyki.

## 2

### Edytor tekstu

## Dokumenty tekstowe. Polskie litery, znaki specjalne, skróty klawiszowe.

Na tej lekcji nauczysz się:

1. uruchamiać wybrany edytor tekstowy;
2. pisać polskie znaki, wielkie litery oraz znaki interpunkcyjne;
3. korzystać z podstawowych klawiszy specjalnych;
4. korzystać z podstawowych skrótów klawiszowych.

### Aplikacje

Edytor tekstowy to aplikacja służąca do pisania tekstów. Dostępne są różne edytory tekstu.

Przykładem bardzo prostego edytora tekstu jest program *Notatnik*, który znajdziesz w akcesoriach systemu.

W tym rozdziale przedstawiamy podstawowe działanie dwóch edytorów tekstu:



LibreOffice Writer  
(czyt.  
libre ofis rajter)


oraz



Microsoft Word  
(czyt.  
majkrosoft łord)

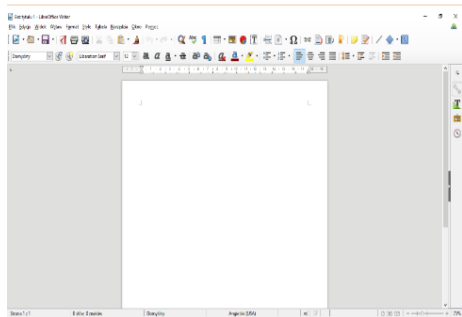
### Uruchamianie programu

Proces uruchomienia programu będzie nieznacznie różnił się w zależności od wersji systemu operacyjnego zainstalowanego na twoim komputerze.

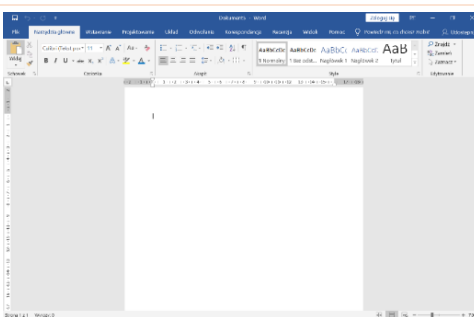
Przykładowo dla systemu Windows 10 możliwe jest znalezienie dowolnego zainstalowanego programu w Menu Start .

## Pusty dokument

LibreOffice Writer



Microsoft Word



### Litery **ą, ć, ę, ł, ń, ó, ś, ź, ż**

Na klawiaturze komputera znajdziesz litery alfabetu łacińskiego. W języku polskim występują litery, które posiadają znaki diakrytyczne, czyli symbole graficzne umieszczane nad lub pod literą oraz wewnątrz niej.

W języku polskim jest dziewięć takich liter: **ą, ć, ę, ł, ń, ó, ś, ź, ż**.

Aby otrzymać je na ekranie komputera, należy skorzystać z następujących kombinacji klawiszowych:

<b>ą</b>	prawy Alt + a	<b>ł</b>	prawy Alt + l	<b>ś</b>	prawy Alt + s
<b>ć</b>	prawy Alt + c	<b>ń</b>	prawy Alt + n	<b>ż</b>	prawy Alt + z
<b>ę</b>	prawy Alt + e	<b>ó</b>	prawy Alt + o	<b>ź</b>	prawy Alt + x

### Wielkie litery

Wielkie litery tworzy się poprzez wciśnięcie klawisza **Shift** (czyt. szifit) wraz z literą na klawiaturze.

Na przykład:

<b>A</b>	Shift + a	<b>B</b>	Shift + b	<b>C</b>	Shift + c
----------	-----------	----------	-----------	----------	-----------

## Wielkie litery ze znakami diakrytycznymi

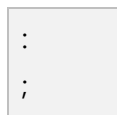
Ą Shift + prawy Alt + a    Ł Shift + prawy Alt + l    Ś Shift + prawy Alt + s  
Ć Shift + prawy Alt + c    Ń Shift + prawy Alt + n    Ź Shift + prawy Alt + z  
Ę Shift + prawy Alt + e    Ó Shift + prawy Alt + o    Ż Shift + prawy Alt + x

## Znaki interpunkcyjne

Znaki interpunkcyjne są niezbędne do poprawnego zapisu tekstu.  
W polskim alfabecie mamy dziesięć znaków interpunkcyjnych. Są to:

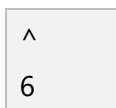
kropka	.	wielokropek	...
przecinek	,	znak zapytania	?
średnik	;	wykrzyknik	!
dwukropek	:	nawias	()
myślnik	-	cudzysłów	„”

Na klawiaturze na wielu klawiszach  
widnieją po dwa symbole, na przykład:



Po wciśnięciu klawisza w edytorze tekstu pojawi się znak widniejący na dole danego klawisza. Jeśli na klawiaturze komputera wciśniesz klawisz umieszczony w powyższym przykładzie, to otrzymasz znak „;” (średnik). Aby otrzymać znak widniejący w górnej części klawisza, w tym przypadku „:” (dwukropek), wciśnij dany klawisz równocześnie z klawiszem Shift.

Naciskając Shift i klawisz



, otrzymasz znak ^ .

Naciskając Shift i klawisz



, otrzymasz znak \* .

## Klawisze specjalne

W poprzednich rozdziałach poznaliśmy już niektóre z klawiszy spełniających specjalne funkcje w programie. Te klawisze noszą nazwę klawiszy specjalnych.

W tabeli poniżej znajdują się przykłady klawiszy specjalnych wraz z ich funkcjami.

Enter	przejdźcie do następnej linii, potwierdzenie operacji	Shift	wielkie litery, znaki interpunkcyjne, kombinacje klawiszowe
Spacja	odstęp w linii między znakami	Backspace (czyt. bakspejs)	usunięcie ostatniego znaku
Alt	polskie litery, kombinacje klawiszowe	Ctrl	kombinacje klawiszowe
Caps Lock (czyt. kaps lok)	ciągłe pisanie wielkimi literami	Tab	duży odstęp w linii tekstu

## Skróty klawiszowe

Skróty klawiszowe to kombinacje klawiszy, których wciśnięcie spowoduje wykonanie operacji na komputerze.

W tabeli poniżej znajdują się przykłady skrótów klawiszowych wraz z ich funkcjami:

Ctrl+C	skopiowanie zaznaczonej części tekstu, dokumentu	Ctrl+V	wklejenie skopiowanej lub wyciętej części tekstu, dokumentu
Ctrl+X	wycięcie zaznaczonej części tekstu, dokumentu	Ctrl+Z	cofnięcie o jedną operację

Co ciekawe, te same skróty klawiszowe można wykorzystać w innych programach. Skróty klawiszowe niejednokrotnie ułatwią ci pracę nie tylko podczas korzystania z edytora tekstu.



## Zadanie 1.

Otwórz nowy dokument w programie LibreOffice Writer lub Microsoft Word.

Przepisz poniższe słowa, korzystając z polskich znaków diakrytycznych:

łązając, łożąc, łożmyż, łożże, łużąc, żąleś, żęleś, żółcą, żółcę, żółć.

## Zadanie 2.

Przepisz poniższe zdania, korzystając z wielkich liter i znaków interpunkcyjnych.

Karolina krzyknęła: Adam, zawołaj Laurę!

Bardzo lubię książkę „Quo vadis” (choć była nieco zbyt długa).

Wojtku, czy nie żal Ci straconego czasu?

## Zadanie 3.

Napisz poniższe zdanie, używając klawisza CapsLock. Następnie, korzystając ze skrótu klawiszowego Ctrl+C – kopiuje i Ctrl+V – wklej, skopiuje je 7 razy w następujący sposób:

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

STÓŁ Z POWYŁAMYWANymi NOGAMI.

Zapisz dokument w swoim folderze, nazywając plik „Dok\_tekst\_Imię\_Nazwisko\_Klasa”.

## Formatowanie czcionek.

## Listy numerowane.

Na tej lekcji nauczysz się:

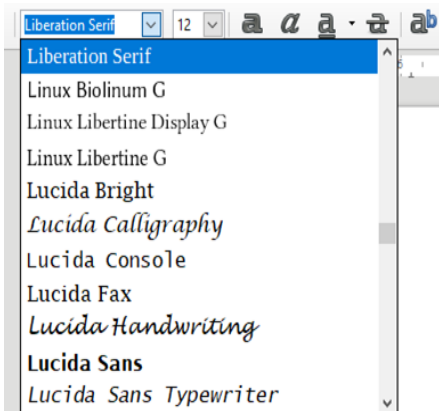
1. pisać i edytować różnorodny tekst;
2. podstawowych zasad pisania i edycji tekstu;
3. tworzyć jednopoziomową listę numerowaną.

### Pisanie i edycja tekstu

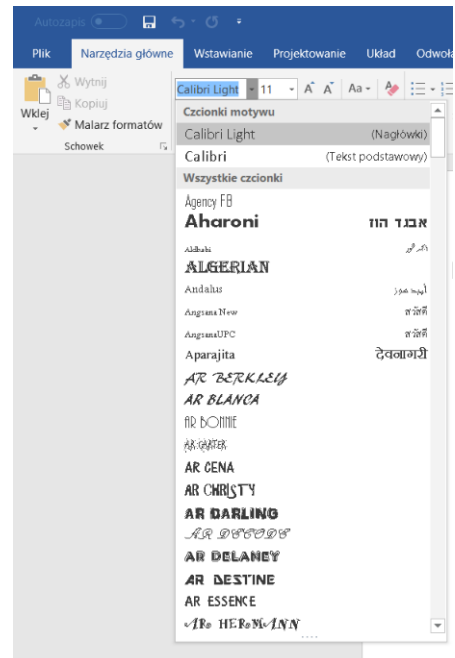
Edytor tekstowy pozwala na wybór i zmianę wielu cech tekstu wpływających na jego wygląd.

Rozmiar i krój czcionki wybiera się w odpowiednim polu, w górnym pasku programu. Czcionkę i rozmiar wybierz w sposób pokazany na rysunku poniżej.

LibreOffice Writer



Microsoft Word



Wybierając rodzaj i krój czcionki, możesz zmienić wygląd tekstu.

Przykład

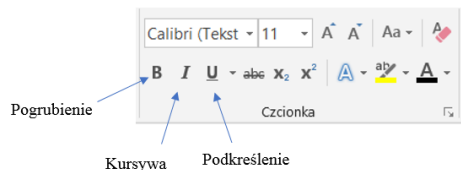
Najdłuższy dzień roku. (Times New Roman, 13)  
Najkrótszy dzień roku. (Comic Sans MS, 15)

**Pogrubienie**, *kursywę*, podkreślenie wybierasz w górnym panelu edytora tekstowego. W ten sposób możesz zmienić formatowanie fragmentu tekstu, na który chcesz zwrócić szczególną uwagę czytelnika.

LibreOffice Writer



Microsoft Word



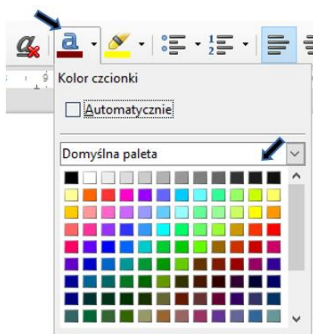
Przykład

**Najdłuższy dzień** roku. (Liberatrion Serif, 12)

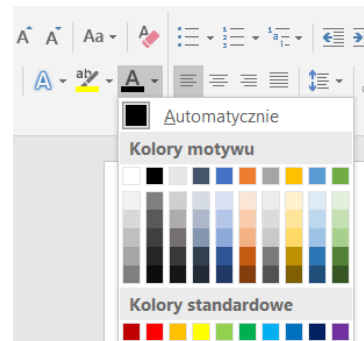
*Najkrótszy* **dzień** roku. (Liberatrion Serif, 12)

Zmiany koloru czcionki dokonuje się poprzez zaznaczenie odpowiedniego pola w górnym pasku programu i wybór koloru.

LibreOffice Writer



Microsoft Word



Przykład

**Najdłuższy dzień roku.** (Liberatrion Serif, 12)

**Najkrótszy dzień roku.** (Liberatrion Serif, 12)

## Edycja istniejącego tekstu

Jeśli chcesz zmienić formatowanie istniejącego tekstu, wystarczy, że zaznaczysz fragment, który chcesz zmienić, a następnie wybierzesz interesujące cię opcje, na przykład kursywę i czerwony kolor tekstu.

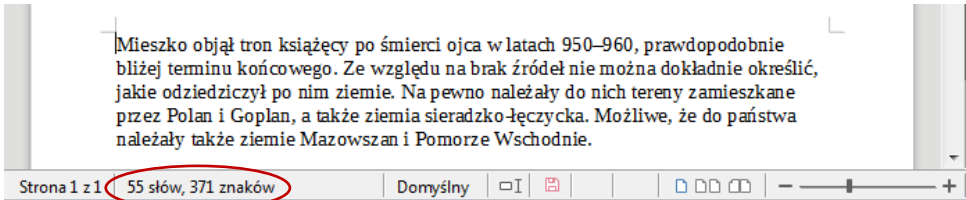
Przykład

Stół z powylamywanymi nogami. W czasie suszy szosa sucha.  
(Agency FB, 12)

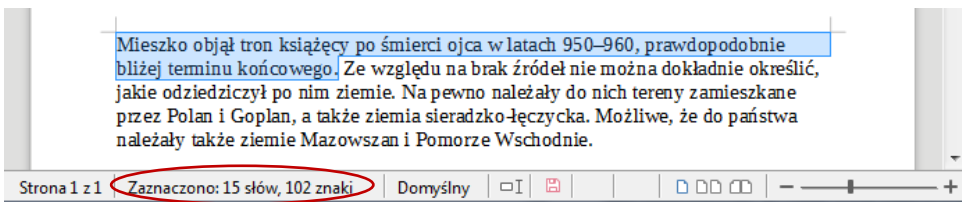
*Stół z powylamywanymi nogami. W czasie suszy szosa sucha.*  
(Agency FB, 12, kursywa, kolor czerwony)

## Liczba słów i znaków

Czasem dobrze jest wiedzieć, z ilu słów i znaków składa się Twój tekst. Informacja ta znajduje się na samym dole okna programu.



Natomiast jeśli chcesz znać liczbę słów i znaków składających się na fragment tekstu, wystarczy zaznaczyć go przytrzymując lewy przycisk myszy.



## Zasady pisania i edycji tekstu

Po zapoznaniu się z podstawowymi narzędziami edytora tekstowego, przyszedł czas na naukę zasad poprawnej pisowni.

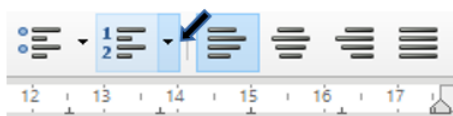
Poniżej zostały przedstawione najważniejsze zasady poprawnej pisowni, do których należy się stosować:

1. Przed znakiem interpunkcyjnym nie stawiaj spacji.
2. Stawiaj spację po znaku interpunkcyjnym.
3. Nie stawiaj spacji między początkiem i końcem tekstu a nawiasami lub cudzysłowem.
4. Nie stosuj więcej niż jednej spacji jako odstępu między słowami.
5. Każde nowe zdanie zaczynaj wielką literą i kończ kropką.
6. Pamiętaj o zasadach ortografii i interpunkcji.

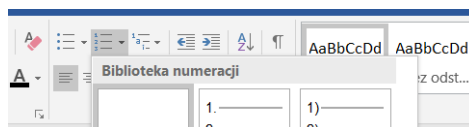
## Listy jednopoziomowe

Niektóre informacje, jak na przykład plan wycieczki szkolnej czy listę zakupów, wygodnie jest zapisać w postaci list numerowanych. W grupie narzędzi *Akapit*, w górnym pasku programu, znajdziesz pole *Numerowanie*.

LibreOffice Writer



Microsoft Word



Przykład użycia listy numerowanej

Plan wycieczki szkolnej:

1. Przyjazd do Krakowa.
2. Zwiedzanie Zamku Królewskiego na Wawelu.
3. Zwiedzanie Sukiennic.
4. Czas na odpoczynek.
5. Zwiedzanie Starego Miasta.
6. Wejście na wieżę kościoła Mariackiego (dla chętnych).
7. Obiad.
8. Powrót do domu.

Przejdźcie do następnego punktu wykonuje się poprzez wciśnięcie klawisza Enter. Jeśli chcesz zakończyć listę i wrócić do pisania zwykłego tekstu, wystarczy dwukrotnie wcisnąć klawisz Enter.

### Zadanie 1.

Otwórz stworzony na ostatnich zajęciach dokument tekstowy.

Przepisz poniższe zdania, korzystając z podanych w nawiasach rodzajów oraz rozmiarów czcionki.

W 2017 roku na **Ziemi** żyło **7,5 miliardów** ludzi. (Arial, 12)

**Najwięcej** ludzi na *świecie* żyje w Chinach, **1,4 miliarda**. (Times New Roman, 14)

W **Polsce** żyje 38 milionów ludzi. (Comic Sans MS, 10)

Piękną **pogodę** *zawdzięczamy* wyżowi znad **Atlantyku**. (Calibri, 16)

## **Zadanie 2.**

Przepisz, wraz z tytułem, poniższą listę numerowaną.

### **Największe kraje świata (pod względem powierzchni):**

1. Rosja
2. Kanada
3. Chiny
4. USA
5. Brazylia
6. Australia
7. Indie
8. Argentyna
9. Kazachstan
10. Algieria

Zapisz swoje postępy.

## **Zadanie 3.**

Napisz w 10 punktach plan wydarzeń opowiadania pt.: „Niezwykły dzień w szkole”, inspirowany książką J.J. Sempego i R. Gosciniego „Mikołajek”.

Musi on zawierać:

1. tytuł napisany czcionką Times New Roman, wielkość 16, pogrubioną, w kolorze niebieskim;
2. listę numerowaną, napisaną czcionką Times New Roman, wielkość 12, kolor czarny.

Przynieś wykonany przez siebie plan na następne zajęcia na przenośnym dysku USB.

## Formatowanie tekstu.

### Akapity, wcięcia, interlinie.

Na tej lekcji nauczysz się:

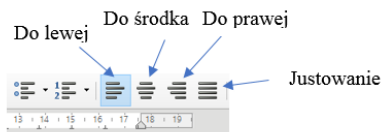
1. wyrównywać tekst na stronie;
2. dzielić tekst na akapity, w tym dodawać wcięcia w tekście oraz odstępy między liniami;
3. formatować tekst;
4. praktycznie stosować nabyte umiejętności.

Estetyka tworzonego tekstu jest istotna. Składają się na nią elementy formatowania czcionki, które zostały poznane na poprzedniej lekcji. Narzędzia edytora tekstowego umożliwiają dodatkowo wyrównanie tekstu oraz wykonanie wcięć i odstępów między liniami.

### Wyrównywanie tekstu

Aby wyrównać fragment tekstu do jednej z krawędzi strony lub do środka, należy zaznaczyć kursorem ten fragment, a następnie wybrać odpowiedni przycisk z karty *Akapit* w górnym pasku programu:

LibreOffice Writer



Microsoft Word



„Lekcje rozpoczynają się o siódmej rano. Nigdzie chyba chłopcy nie uczą się tak chętnie, jak w Akademii pana Kleksa. Przede wszystkim nigdy nie wiadomo, co pan Kleks danego dnia wymyśli, a po wtóre – wszystko, czego się uczymy, jest ogromnie ciekawe i zabawne.”





### Wyrównanie do środka

„Doktor Paj-Chi-Wo odkrył mi również inne swoje tajemnice oraz nauczył mnie wszystkiego, co dzisiaj umiem. Między innymi wyjawiał mi ukryte znaczenie ludzkich imion. Tym więc tłumaczy się, że do mojej Akademii przyjmuję tylko uczniów, których imiona zaczynają się na literę A, gdyż wiadomo z góry, że są zdolni i pracowici.”



### Wyrównanie do prawej

„Myślę, że sam pan Kleks powstał z takiego właśnie rozgniecionego atramentowego kleksa i dlatego tak się nazywa.”

Opcja justowania umożliwia wyrównanie tekstu jednocześnie do lewej i prawej krawędzi. Dzięki temu tekst na stronie wygląda bardziej estetycznie.



### Justowanie

„Pan Kleks przyjmuje do swojej Akademii tylko tych chłopców, których imiona zaczynają się na literę A, bo – jak powiada – nie ma zamiaru zaśmieszać sobie głowy wszystkimi literami alfabetu.”

*[Akademia Pana Kleksa, Jan Brzechwa]*

## Akapity

Akapit jest podstawowym sposobem dzielenia tekstu na mniejsze fragmenty obejmujące spójne logiczne myśli. Dzięki akapitom tekst staje się czytelniejszy.

Dwa różne akapity rozdzielamy poprzez:

- wcięcie akapitowe, które można dodać, wciskając klawisz Tab,
- odstęp między akapitami, który można dodać poprzez dwukrotne wciśnięcie klawisza Enter na końcu akapitu (opcja niezalecana).

Poniżej zamieszczono przykład fragmentu tekstu składającego się z dwóch akapitów:

3 marca 1969 roku w ramach misji Apollo 9, na orbicie okołoziemskiej przetestowano moduł księżycowy LM, który miał posłużyć do lądowania na Księżycu. 18 maja w ramach misji Apollo 10 moduł księżycowy zbliżył się do powierzchni Księżycza na 14 km, potwierdzając możliwość przeprowadzenia manewru lądowania.

16 lipca 1969 roku rozpoczęła się najważniejsza misja programu – Apollo 11. Członkowie załogi misji – Neil Armstrong, Edwin Buzz Aldrin i Michael Collins - mieli przejść do historii.

Powyższy przykład demonstruje akapity z wcięciami. Akapity możemy też oddzielić za pomocą klawisza Enter.

## **Zasady poprawnego formatowania tekstu**

Oto kilka zasad, dzięki którym zwiększysz przejrzystość swojego tekstu:

1. Wyróżniaj akapity w tekście przez wcięcie akapitowe lub odstępy akapitowe.
2. Pisząc tekst, postaraj się, żeby był on jak najbardziej czytelny. Unikaj używania różnych czcionek i różnych ich wielkości.
3. Unikaj pisania wielkimi literami całych słów; są „krzykiem” w tekście.
4. Dbaj o zasady wyrównywania tekstu:
  - do lewej: przy listach numerowanych i punktowych,,
  - do środka: dla tytułów, nagłóweków,
  - justowanie: dla zasadniczego tekstu.

## Zadanie 1.

Otwórz stworzony na poprzednich zajęciach dokument tekstowy. Przepisz poniższy tekst w dokładnie takiej samej formie.

(Times New Roman, wielkość 12)

**Jan Brzechwa**

### **Atrament i kreda (fragment)**

Wzdychała kreda: „Wciąż jestem biała,

Nie chcę być biała!...” No i-szczerbiała.

Jęczał atrament: „O, losie marny,

Wciąż jestem czarny, kompletnie czarny,

Jak gdyby we mnie kto smołę przelał.

Nie chcę być czarny! Dość już!” I zbiałał.

## Zadanie 2.

Przepisz poniższy tekst, zachowując podział na akapity.

Zastosuj justowanie. (Arial, wielkość 12)

Dlatego też, jeżeli któryś z uczniów wyróżni się podczas lekcji, pan Kleks uroczyście wyjmując z tabakierki świeżą, nie używaną jeszcze piegę i przytwierdza ją do nosa takiego szczęściarza mówiąc:

– Noś ją godnie, mój chłopcze, i nigdy jej nie zdejmuj, jest to bowiem najwyższa odznaka, jaką możesz sobie zdobyć w mojej Akademii.<sup>1</sup>

Pora, by wykorzystać w praktyce poznane umiejętności. Zrealizuj zaproponowany projekt lub inny zdefiniowany przez twojego nauczyciela.

---

<sup>1</sup> *Akademia Pana Kleksa, Jan Brzechwa*

## Projekt międzyprzedmiotowy 3

### Edytor tekstu

### Informatyka / język polski

### Opowiadanie „Niezwykły dzień”

Projekt ten pozwoli ci na wypróbowanie sił w roli pisarza. Twoim zadaniem jest napisanie, w wybranym edytorze tekstowym, opowiadania pod tytułem: „Niezwykły dzień”. Daj się ponieść wyobraźni i wymyśl historię o jakiej nikt nigdy nie słyszał!

Najpierw stwórz plan opowiadania. Na jego podstawie napisz opowiadanie do 1500 znaków, korzystając z wybranego edytora tekstu. Poniżej zostały wyszczególnione elementy, które powinno ono zawierać:

1. Tytuł:
  - wyśrodkowany,
  - czcionka Times New Roman, wielkość 15,
  - kolor niebieski,
  - pogrubiona.
2. Minimum cztery wyjustowane akapity stworzone z zastosowaniem poznanych zasad:
  - wstęp (może zawierać jeden lub więcej akapitów),
  - rozwinięcie (co najmniej dwa akapity),
  - zakończenie (co najmniej jeden akapit),
3. Tekst napisany czcionką Times New Roman, wielkość 12.
4. Twój podpis i datę napisane kursywą i wyrównane do prawej, umieszczone na końcu opowiadania.

Wykorzystaj narzędzia poznane na lekcji. Możesz wybrać jedną z dwóch aplikacji omówionych w podręczniku lub skorzystać z innej. Jak zapewne zauważysz, aplikacje podobnego typu mają podobne funkcje i podobnie się z nich korzysta.

## Projekt międzyprzedmiotowy 4

### Edytor tekstu

#### Informatyka / język polski

#### Opowiadanie „Niesamowita historia”

Kolejną propozycją projektu jest opowiadanie pod tytułem „Niezwykła historia”. Tym jednak razem będzie ono osadzone na tle wybranego, ważnego wydarzenia z historii Polski. Opisz to co zobaczyłeś / zobaczyłaś. Wyobrażenia tu nie wystarczy, twoja wiedza historyczna również będzie miała znaczenie!

Najpierw stwórz plan opowiadania. Na jego podstawie napisz opowiadanie do 1500 znaków, korzystając z wybranego edytora tekstu. Poniżej zostały wymienione elementy, które powinno ono zawierać:

1. Tytuł:
  - wyśrodkowany,
  - czcionka Times New Roman, wielkość 15,
  - kolor niebieski,
  - pogrubiona.
2. Minimum cztery wyjustowane akapity stworzone z zastosowaniem poznanych zasad:
  - wstęp (może zawierać jeden lub więcej akapitów),
  - rozwinięcie (co najmniej dwa akapity),
  - zakończenie (co najmniej jeden akapit),
3. Tekst napisany czcionką Times New Roman, wielkość 12.
4. Twój podpis i datę napisane kursywą i wyrównane do prawej, umieszczone na końcu opowiadania.

Wykorzystaj narzędzia poznane na lekcji. Możesz wybrać jedną z dwóch aplikacji omówionych w podręczniku lub skorzystać z innej.

Jak zapewne zauważysz, aplikacje podobnego typu mają podobne funkcje i podobnie się z nich korzysta.

## 3

# Prezentacje

## Dodawanie i usuwanie slajdu.

## Wstawianie tekstu

Na tej lekcji nauczysz się:

1. uruchamiać program do tworzenia prezentacji multimedialnych;
2. dodawać, edytować i usuwać slajdy;
3. dodawać tekst do slajdu.

## Aplikacje

Dostępne są różne aplikacje do tworzenia prezentacji. Wszystkie mają podobne funkcje. W tym rozdziale przedstawiamy podstawowe działanie dwóch z nich:

### **LibreOffice Impress**

(czyt. libre ofis impres)      oraz




### **Microsoft PowerPoint**

(czyt. majkrosoft paer pojnt)

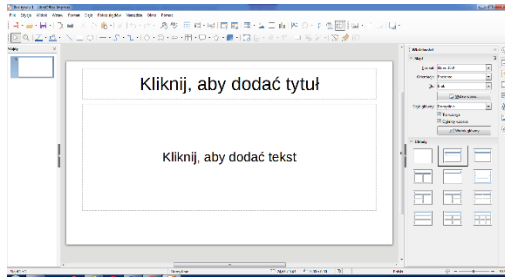


## Uruchamianie programu

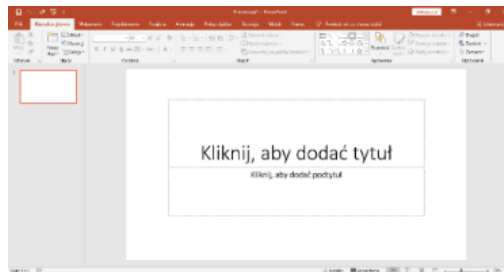
Proces uruchomienia programu będzie nieznacznie różnił się w zależności od wersji systemu operacyjnego zainstalowanego na twoim komputerze. Przykładowo dla systemu Windows 10 możliwe jest znalezienie dowolnego zainstalowanego programu w Menu Start .

## Pusty projekt

- LibreOffice Impress



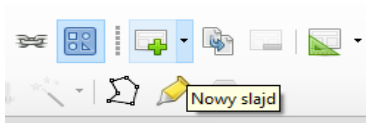
- Microsoft PowerPoint



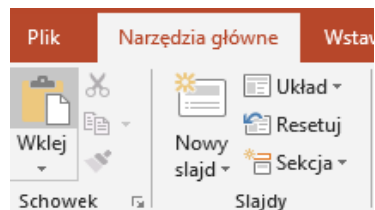
## Dodawanie slajdów

W lewym górnym rogu aplikacji na karcie *Narzędzia główne* znajdziesz miniaturę slajdu podpisaną *Nowy slajd*. Aby dodać nowy slajd, kliknij na tę pozycję.

LibreOffice Impress

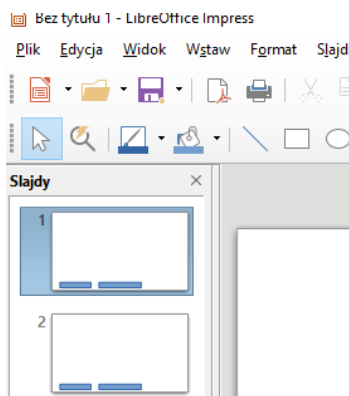


Microsoft PowerPoint

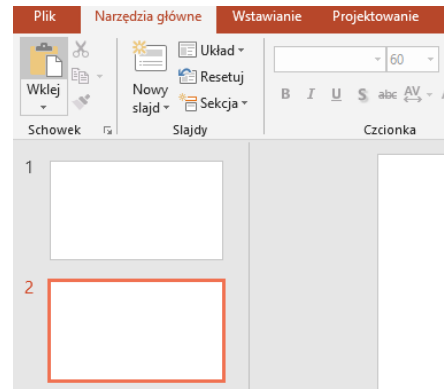


Nowe slajdy będą pojawiać się w zakładce po lewej stronie ekranu. Między różnymi slajdami przechodzi się, klikając na miniaturkę wybranego slajdu na liście.

## LibreOffice Impress



## Microsoft PowerPoint



## Usuwanie slajdów

Co zrobić w sytuacji, gdy slajd nie jest już potrzebny? Operacja usuwania slajdów jest równie prosta jak ich tworzenie. Wystarczy zaznaczyć wybrany slajd i wcisnąć klawisz *Delete*.

## Zmiana tła

Dobrze dobrany kolor tła slajdów sprawi, że twoja prezentacja będzie wyglądała atrakcyjnie.

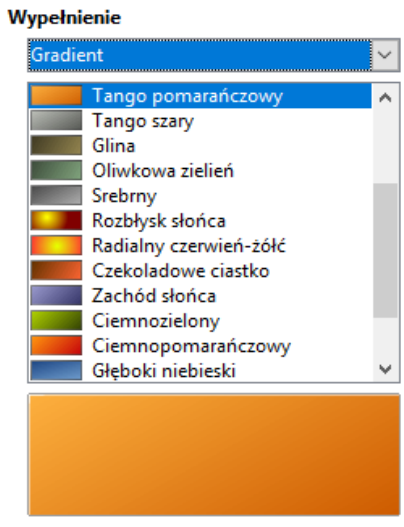


Aby wybrać tło:

### LibreOffice Impress

1. Kliknij prawym przyciskiem myszy w dowolnym punkcie slajdu.
2. Spośród możliwych opcji wybierz *Właściwości slajdu*.
3. Gdy pojawi się panel *Ustawienia strony*, wybierz zakładkę *Tło*.
4. W zakładce *Tło* wybierz swoje wypełnienie slajdu.

Przykładowo wybierz rodzaj wypełnienia Gradient i kolor Tango pomarańczowy.

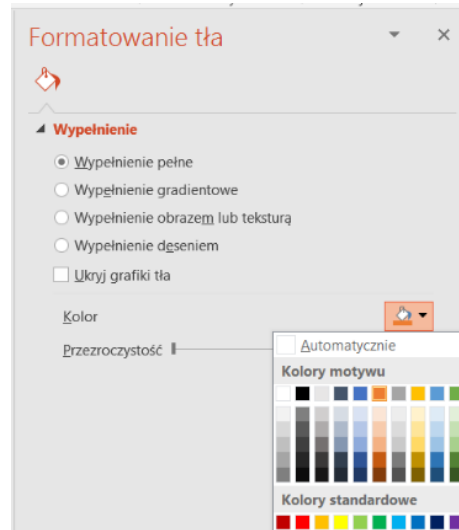


### Microsoft PowerPoint

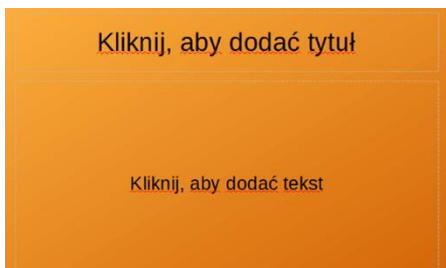
1. W górnym panelu na karcie *Projektowanie* w grupie *Dostosowywanie* kliknij w opcję *Formatowanie tła*.
2. Wybierz wypełnienie pełne oraz wybrany kolor.

Przykładowo wybierz *Wypełnienie pełne* i kolor pomarańczowy.

Aby tło zostało zmienione dla wszystkich slajdów, zaznacz opcję *Zastosuj do wszystkich*.



Slajd z nowym tłem będzie wyglądał następująco:



## Dodawanie tekstu do slajdu

Potrafisz już dodawać i usuwać slajdy oraz zmieniać tło. Głównym zadaniem prezentacji jest przekazanie odbiorcom pewnych informacji. Czas nauczyć się, w jaki sposób dodać tekst.

- Aby dodać tytuł do slajdu, wystarczy postąpić zgodnie z instrukcją programu. Kliknij w pole *Kliknij, aby dodać tytuł* i zacznij pisać.
- Aby dodać treść do slajdu, kliknij w pole *Kliknij, aby dodać podtytuł* i zacznij pisać.

Zwróć uwagę, że podczas pisania tekstu, w górnym panelu dostępne są, znane z edytora tekstu, opcje formatowania. Wielkość i rodzaj czcionki ustawione są domyślnie. Jeśli chcesz, możesz je zmienić.

## Zadanie 1.

Otwórz nowy projekt w programie do prezentacji. Dodaj dwa nowe slajdy do prezentacji, a następnie zmień kolor tła na niebieski.

## Zadanie 2.

Do każdego ze slajdów dodaj zaproponowane poniżej tytuły i treści zachowując domyślną czcionkę.

- Slajd 1 (tytułowy)
  - Tytuł: Ziemia
  - Tekst: Nasze miejsce w Kosmosie
- Slajd 2
  - Tytuł: Podstawowe informacje
  - Tekst:
    - Planeta skalista
    - Wiek: 4,5 miliarda lat
    - Okres obiegu wokół Słońca: 365,25 dni
    - Obwód równika: 40 075 kilometrów
- Slajd 3
  - Tytuł: Ziemia w Układzie Słonecznym
  - Tekst:
    - Trzecia planeta od Słońca
    - Największa planeta skalista
    - Odległość od Słońca: 150 mln kilometrów
    - Jedyna „żyjąca” planeta

Zapisz stworzoną przez siebie prezentację w swoim folderze pod nazwą „Prezentacja1\_imię\_nazwisko\_klasa”.

## Wklejanie ilustracji.

### Zasady tworzenia prezentacji.

### Tryb prezentacji

Na tej lekcji nauczysz się:

1. dodawać ilustracje do slajdów;
2. zasad tworzenia prezentacji;
3. uruchomić tryb prezentacji;
4. stosować w praktyce nabyte umiejętności.

### **Dodawanie ilustracji**

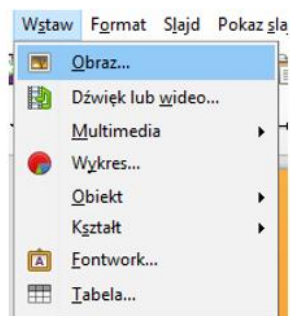
Wzbogacenie prezentacji o ilustracje zwiększa jej atrakcyjność. Ilustracjami mogą być twoje własne zdjęcia lub rysunki stworzone w programie graficznym. Można również skorzystać ze zdjęć dostępnych w internecie. Należy pamiętać, by nie naruszyć praw autorskich właścicieli zdjęć. Jeśli zdjęcie zostało rozpowszechnione za zgodą twórcy dzieła, to można wykorzystać je w celach edukacyjnych (na przykład użyć ich w prezentacji w ramach zajęć szkolnych). Za każdym razem gdy wykorzystujemy cudze zdjęcie, musimy podać źródło (na przykład stronę internetową, na której znaleźliśmy zdjęcie) oraz autora (jeśli jesteśmy w stanie go ustalić).

Nauczymy się teraz, jak zapisać udostępnione w internecie zdjęcie, które chcemy umieścić w naszej prezentacji. Szczegóły tej procedury mogą się nieco różnić w zależności od strony internetowej. Wybierz stronę i wejdź. Najedź kursorem na wybrane zdjęcie i kliknij lewym przyciskiem myszy. Zdjęcie powinno wypełniać teraz centralną część ekranu. Aby zapisać je w pamięci komputera, wystarczy kliknąć teraz na jego obszarze prawym przyciskiem myszy, wybrać „Zapisz obraz jako” i następnie wybrać odpowiedni folder i nazwę. Niektóre witryny oferują zapis zdjęcia poprzez wybranie odpowiedniego przycisku, na przykład „Pobierz oryginalny plik”.

Gdy ilustracja znajduje się na dysku naszego komputera, możemy dodać ją do prezentacji.

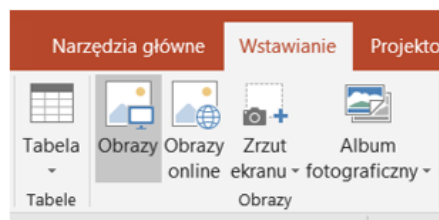
## LibreOffice Impress

1. W programie prezentacji wybierz zakładkę *Wstaw*, znajdującą się w górnym panelu, a następnie zaznacz opcję *Obraz*.
2. Na ekranie pojawi się okno *Wstaw obraz*. Musisz teraz odszukać folder, w którym zapisane jest zdjęcie, którego chcesz użyć. Gdy je odnajdziesz, zaznacz opcję *Otwórz*.



## Microsoft PowerPoint

1. W programie prezentacji wybierz zakładkę *Wstawianie* znajdującą się w górnym panelu, a następnie zaznacz opcję *Obrazy*.
2. Na ekranie pojawi się okno *Wstaw obraz*. Musisz teraz odszukać folder, w którym zapisane jest zdjęcie, którego chcesz użyć. Gdy je odnajdziesz, zaznacz opcję *Otwórz*.



W obu przypadkach slajd będzie wyglądał podobnie.

Zmiana rozmiaru oraz położenia obrazka odbywa się tak samo jak w przypadku zaznaczenia dla programu graficznego.



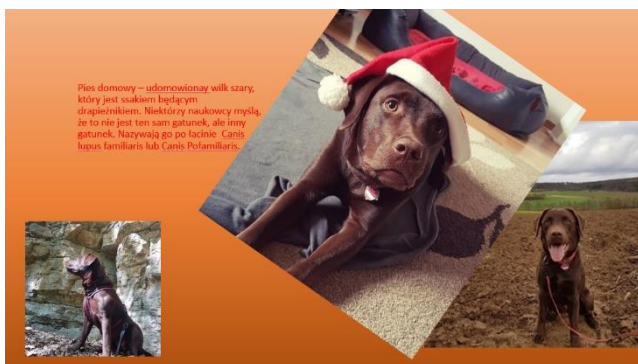
Źródło i/lub autora zdjęcia możesz uwzględnić na przykład poprzez dodanie odpowiedniej adnotacji pod zdjęciem.

## Podstawowe zasady tworzenia dobrych prezentacji

Nawet najlepiej przygotowana pod względem treści prezentacja może nie wzbudzić zainteresowania odbiorców, jeśli slajdy zostały stworzone bez pomysłu lub nieestetycznie. Aby prezentacja była łatwa w odbiorze, a do tego czytelna i estetyczna, należy przestrzegać poniższych zasad.

1. Zanim zaczniesz tworzyć prezentację, zaplanuj co i w jaki sposób chcesz przekazać.
2. Wykorzystaj slajd tytułowy, żeby przedstawić odbiorcom główną myśl.
3. Staraj się, by twoje slajdy były jak najbardziej czytelne. Nie umieszczaj dużej ilości tekstu.
4. Zamieszczaj tylko istotne informacje w postaci jak najkrótszych haseł.
5. Każdy slajd powinien mieć osobny tytuł.
6. Konsekwentnie stosuj kolory, użyj jedynie kilku.
7. Stosuj czytelną czcionkę, bez zbędnych ozdób.
8. Używaj dużych rozmiarów czcionek, by można było odczytać tekst z daleka.
9. Ilustracje powinny być czytelne i dobrze widoczne.

Oto przykład niepoprawnie stworzonego slajdu:



Oto kilka błędów, które popełnił twórca:

- brak tytułu slajdu;
- mało czytelny tekst (za mała czcionka, kolor czcionki źle dobrany do koloru tła);
- tekst jest napisany pełnymi zdaniami, zamiast krótkimi hasłami;
- nieczytelne ilustracje (małe rozmiary zdjęć, zbyt duża liczba zdjęć),
- brak autora i/lub źródła zdjęć.

## **Zasady prezentowania**

Aby twoja prezentacja została dobrze odebrana przez słuchaczy, nie wystarczy, że przygotujesz ją poprawnie. Musisz również umieć ją przedstawić.

Oto kilka zasad, które sprawią, że twoje prezentacje zostaną lepiej odebrane.

1. Prezentuj z pamięci. Nie czytaj tekstu z prezentacji. Twoja prezentacja jest tłem i ilustracją twojej wypowiedzi, nie jej treścią.
2. Bądź zwrócony twarzą do klasy. Stój prosto. Mów do koleżanek i kolegów, starając się zmieniać osoby, na które patrzysz.
3. Zwróć uwagę na swój głos. Nie możesz mówić zbyt cicho ani zbyt monotonicznie.
4. Bądź przygotowany. Przećwicz swoje wystąpienie w domu.

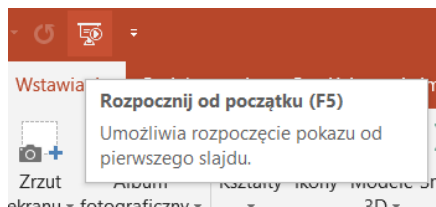
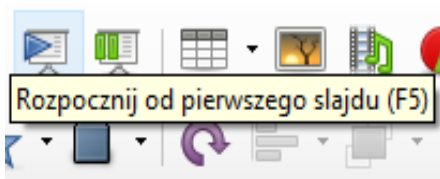
## **Tryb prezentacji**

Tryb prezentacji to wygodny sposób na przedstawienie gotowej prezentacji. W tym trybie prezentacja wyświetlana jest slajd po slajdzie.

Aby uruchomić tryb prezentacji, wystarczy wybrać go na górnym pasku lub wcisnąć klawisz F5.

LibreOffice Impress

Microsoft PowerPoint



Aby przemieszczać się między slajdami, użyj lewego przycisku myszy (następny slajd), prawego przycisku myszy (poprzedni slajd), kółka myszy lub klawiszy strzałek.

## Zadanie 1.

Otwórz stworzoną podczas ostatnich zajęć prezentację, a następnie:

1. Znajdź w sieci zdjęcie Ziemi i dodaj je do tytułowego slajdu.

Wejdz na przykład na stronę:

<https://www.nasa.gov/sites/default/files/thumbnails/image/abifulldiscindex.jpg> i wykorzystaj zdjęcie Ziemi, dodając je do tytułowego slajdu.

2. Znajdź w sieci zdjęcie Ziemi z widoczną satelitą i dodaj je do drugiego slajdu.

3. Znajdź w sieci grafikę ilustrującą Układ Słoneczny i dodaj ją do trzeciego slajdu.

Wejdz na przykład na stronę:

[https://www.jpl.nasa.gov/spaceimages/images/largesize/PIA12114\\_hires.jpg](https://www.jpl.nasa.gov/spaceimages/images/largesize/PIA12114_hires.jpg) i wykorzystaj grafikę obrazującą Układ Słoneczny, dodając ją do trzeciego slajdu.

Pamiętaj o dodaniu źródeł i/lub autorów zdjęć. Zapisz prezentację.

Pora, by wykorzystać w praktyce poznane umiejętności. Zrealizuj projekt zaproponowany w tym rozdziale lub inny zdefiniowany przez twojego nauczyciela.



## Projekt międzyprzedmiotowy 5

### Prezentacje

#### Informatyka / historia „Władca Polski”

Pamiętacie projekt z języka polskiego, w którym mieliście przenieść się w czasie? Jeśli tak, to macie już wprawę. Jeśli nie, to nadarza się okazja, by tego doświadczyć. Ruszacie w przeszłość. Tym razem jednak nie będziecie sami! W podróż zabierzecie ze sobą kolegów i koleżanki z klasy, z którymi będziecie współpracować. Dzięki temu doświadczycie pracy w grupie! Waszym celem będzie przygotowanie prezentacji multimedialnej na temat wybranego władcy Polski i przedstawienie jej na forum klasy.

Dobierzcie się w grupy trzyosobowe. Ustalcie między sobą rok, w który się przenosicie. Sprawdźcie w dostępnych źródłach, kto władał Polską w tym czasie. Waszym zadaniem jest stworzenie prezentacji multimedialnej na temat wybranego władcy oraz przygotowanie 5-minutowego wystąpienia na jej podstawie. Rozdzielcie zadania tak, żeby każde z was miało swój udział podczas wystąpienia.

Właściwie wykonana prezentacja powinna zawierać:

- wybrane przez siebie tło,
- 6 slajdów:
  - slajd tytułowy,
  - dane biograficzne,
  - informacje o rodzinie,
  - ważne wydarzenia, które miały miejsce za panowania władcy,
  - najważniejsze osiągnięcia,
  - podsumowanie postaci.
- co najmniej 3 grafiki lub zdjęcia.

Do zebrania informacji potrzebnych do wykonania zadania wykorzystajcie źródła dostępne w internecie. Przynieście gotowy projekt na dysku USB i zaprezentujcie wspólnie swoją pracę na lekcji historii.

Wykorzystajcie narzędzia poznane na lekcji. Możecie wybrać jedną z dwóch aplikacji omówionych w podręczniku lub skorzystać z innej. Jak zapewne zauważycie, aplikacje podobnego typu mają podobne funkcje i podobnie się z nich korzysta.

## 4

# Arkusz kalkulacyjny

## Komórki, kolumny, wiersze i ich formatowanie.

Na tej lekcji dowiesz się:

1. do czego służy arkusz kalkulacyjny;
2. jak uruchomić arkusz kalkulacyjny;
3. czym są komórki, kolumny oraz wiersze i jak je oznaczać;
4. jak formatować komórki, kolumny i wiersze.

Wiesz już, że programy graficzne służą do tworzenia rysunków, a edytory tekstowe umożliwiają wygodne pisanie oraz formatowanie tekstów. Zajmiemy się teraz kolejnym typem niezwykle przydatnych programów - arkuszami kalkulacyjnymi. Umożliwiają one między innymi:

- łatwy i czytelny zapis informacji i danych (głównie liczbowych) w postaci kolumn i wierszy,
- wykonywanie operacji na danych,
- prezentowanie danych w formie wykresów.

### Aplikacje

Arkusz kalkulacyjny to aplikacja służąca do operacji na liczbach. Dostępne są różne arkusze kalkulacyjne, niemniej wszystkie mają podobne podstawowe funkcje.

W tym rozdziale przedstawiamy podstawowe działanie dwóch arkuszy kalkulacyjnych:

LibreOffice Calc  
(czyt. libre ofis kalk)




oraz

Microsoft Excel (czyt.  
majkrosoft eksel)

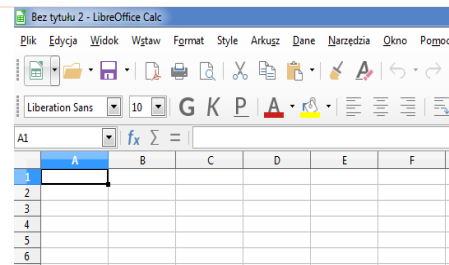


## Uruchamianie programu

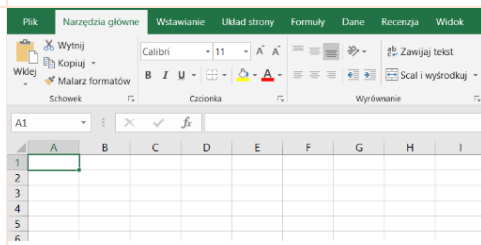
Proces uruchomienia programu będzie nieznacznie różnił się w zależności od wersji systemu operacyjnego zainstalowanego na twoim komputerze. Przykładowo dla systemu Windows 10 możliwe jest znalezienie dowolnego zainstalowanego programu w Menu Start .

## Nowy arkusz

### LibreOffice Calc

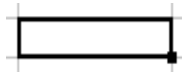


### Microsoft Excel



## Komórki, kolumny, wiersze

Komórka to podstawowa jednostka arkusza. W komórce możemy umieszczać liczby, napisy oraz formuły obliczeniowe.



Kolumna to pionowy zbiór komórek znajdujących się „w jednej linii”. Kolumny oznaczane są literami. Oto przykład kolumny D.



Wiersz to poziomy zbiór komórek znajdujących się „w jednej linii”. Wiersze oznaczane są liczbami. Poniżej przykład wiersza numer 7.



Każda komórka ma swoje oznaczenie, czyli identyfikator. Jest nim litera kolumny i numer wiersza, w których się znajduje. Poniżej zaznaczona jest komórka w kolumnie C i wierszu 5 - w skrócie adres tej komórki to C5.

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				

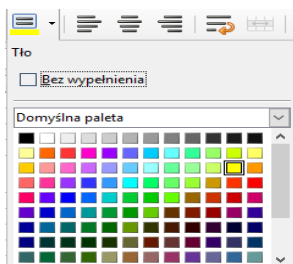
## Formatowanie komórek, kolumn i wierszy

Arkusz kalkulacyjny umożliwia zmianę tła, obramowania oraz czcionki pojedynczej komórki lub kilku zaznaczonych komórek na raz.

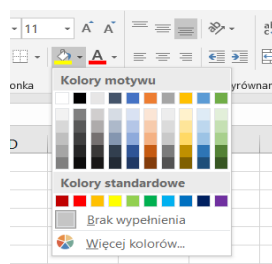
### Zmiana tła

Aby zmienić tło pojedynczej komórki, należy ją zaznaczyć, klikając na nią lewym przyciskiem myszy, a następnie wybrać narzędzie zmiany koloru w górnym pasku programu:

LibreOffice Calc



Microsoft Excel



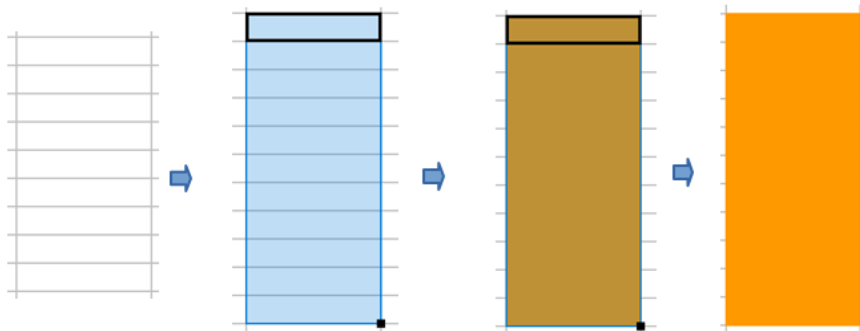
Efekt zmiany koloru komórki:



W ten sam sposób zmienia się kolor większego zbioru komórek. Należy jedynie zaznaczyć kursorem te komórki, których tło chcesz zmienić.

Aby to zrobić, wystarczy przytrzymać lewy przycisk myszy, a następnie przemieszczając myszką, wybrać zbiór komórek, który cię interesuje.

Efekt zmiany koloru zbioru komórek:

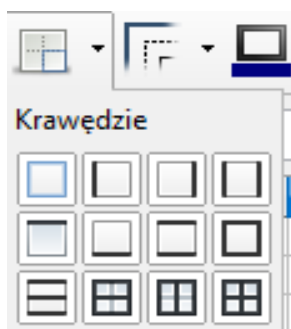


## Obramowanie

Obramowanie komórki lub zbioru komórek dodaje się poprzez ich zaznaczenie. Następnie, w zależności od używanego programu, można podjąć odpowiednie działania.

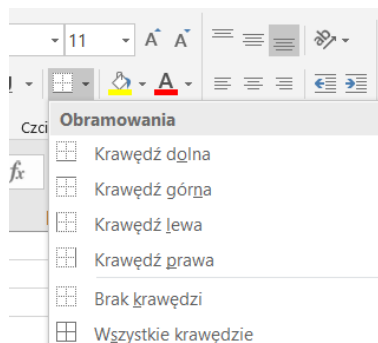
### LibreOffice Calc

Wybierz narzędzie *Krawędzie* znajdujące się w górnym pasku.

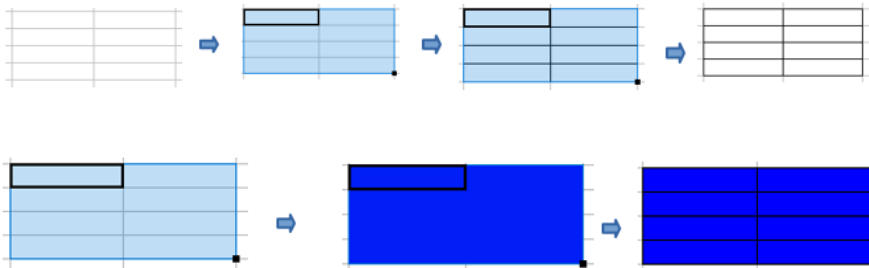


### Microsoft Excel

Na karcie *Narzędzia główne* w grupie *Czcionka* kliknij na pozycję *Obramowania*.




Wybierz opcję *Wszystkie Krawędzie*. Cały proces wygląda następująco:



## Zapisywanie pliku

Dobrą praktyką jest częste zapisywanie postępów swojej pracy. Warto zapisywać plik nie tylko po skończeniu zadania, ale też w trakcie jego wykonywania. W ten sposób uchronisz się przed utratą swojej pracy w przypadku wystąpienia niepożądanych okoliczności losowych np. utraty prądu.

Zapisanie pliku odbywa się dokładnie w ten sam sposób, co w przypadku dokumentów tekstowych czy graficznych:

1. Wybierz w górnym pasku programu symbol  dyskietki.
2. Wybierz folder, w którym chcesz zapisać swoją pracę.
3. Nazwij swój plik i kliknij pozycję *Zapisz*.

Gdy raz zapiszemy plik, następnym razem wystarczy jedynie kliknąć symbol dyskietki, aby zapisać postępy w pracy.

### Zadanie 1.

Zapisz w zeszytcie do informatyki adresy (literę kolumny i numer wiersza) pokolorowanych komórek:

1. żółtej
2. niebieskiej,
3. czerwonej,
4. zielonej.

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

Otwórz nowy arkusz, a następnie wykonaj w nim poniższe zadania:

### Zadanie 2.

Pokoloruj komórki o podanych adresach na dane kolory:

1. zielony: A 7,
2. czerwony: C 9,
3. żółty: B 12,
4. niebieski: D 2.

### Zadanie 3.

Dodaj obramowanie do następujących komórek:

1. A 5
2. B 7
3. D 12
4. C 3

### Zadanie 4.

Pokoloruj na kolor zielony kolumnę F od wiersza 1 do 15.



**Zadanie 5.**

Pokoloruj na kolor czerwony wiersz 14 od kolumny A do E.

**Zadanie 6.**

Dodaj obramowanie do kolumny G od wiersza 1 do 12.

**Zadanie 7.**

Dodaj obramowanie do wiersza 16 od kolumny A do H.

Zapisz stworzony przez siebie arkusz w swoim folderze pod nazwą „Arkusz1\_Imię\_Nazwisko\_klasa”.

## Liczby w komórkach.

### Operacje dodawania, odejmowania, dzielenia i mnożenia wartości dwóch komórek.

Na tej lekcji nauczysz się:

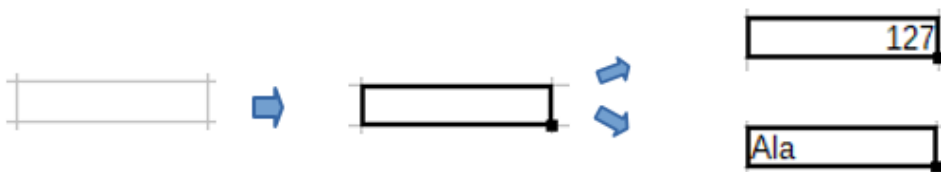
1. wpisywać wartości do komórek w arkuszu;
2. dodawać lub odejmować wartości dwóch komórek;
3. mnożyć lub dzielić wartości dwóch komórek;
4. usuwać zawartość wypełnionej komórki;
5. czym jest pasek formuły i jak z niego korzystać.

Wiesz już, jak nawigować po arkuszu oraz w jaki sposób zmieniać jego wygląd. Najwyższy czas, by nauczyć się, jak wykonywać proste operacje na danych.

### Wypełnianie arkusza

Aby wpisać wartości liczbowe lub litery do komórki, należy ją zaznaczyć, klikając na nią lewym przyciskiem myszy, a następnie wpisać wybraną wartość i nacisnąć klawisz Enter.

Oto przykłady wypełnienia komórek:



Zwróć uwagę na istotną różnicę między tymi dwoma komórkami. Są one przykładami dwóch różnych typów komórek:

Komórki wypełnione samymi liczbami nazywamy komórkami liczbowymi. Można na nich wykonywać operacje matematyczne, o czym zaraz dowiesz się nieco więcej. Przykładem takiej komórki może być:

127

Komórki wypełnione samymi literami bądź literami i liczbami nazywamy komórkami tekstowymi. Nie można na nich wykonywać operacji arytmetycznych. Przykładem takiej komórki może być:

Ala

Zawartość w komórkach tekstowych jest zawsze wyrównana do jej lewej krawędzi, natomiast w komórkach liczbowych do prawej. Warto zwrócić teraz uwagę na jeden istotny szczegół.

Porównajmy ze sobą następujące komórki:

12,4

12a

W tej komórce zawartość jest wyrównana do prawej strony. Oznacza to, że komórka ta jest komórką liczbową.

W tej komórce zawartość jest wyrównana do lewej krawędzi. Oznacza to, że komórka ta jest komórką tekstową.

Zauważ, że program traktuje przecinek jako część liczby.

## Operacje matematyczne

Jak zostało już wspomniane, na komórkach można wykonywać operacje matematyczne.

### Dodawanie

	A	B	C
1	5	12	
2			

Mamy dane dwie liczby w dwóch komórkach. W jaki sposób je do siebie dodać?

1. Zaznacz komórkę, w której chcesz zapisać wynik działania, klikając w nią myszką. W przykładzie będzie to komórka C1:

	A	B	C
1	5	12	

2. Wpisz do zaznaczonej komórki znak równości „=”:

	A	B	C
1	5	12	=

3. Następnie, zaznacz pierwszą z komórek, którą chcesz dodać (w przykładzie jest to komórka A1 z wartością 5). Jej adres pojawi się wtedy w naszej komórce w następujący sposób:

	A	B	C
1	5	12	=A1

4. Jeśli chcemy dodać do siebie te wartości, następnym krokiem jest wpisanie do komórki znaku działania, czyli dla dodawania będzie to znak plus „+”:

	A	B	C
1	5	12	=A1+

5. W ostatnim kroku, należy wybrać drugą komórkę, którą chcemy dodać (w przykładzie jest to komórka B1) i wcisnąć **Enter**, aby potwierdzić działanie:

	A	B	C
1	5	12	=A1+B1

 → 

	A	B	C
1	5	12	17

Otrzymaliśmy poprawny wynik, gdyż  $5 + 12 = 17$ .

## Odejmowanie

Przy odejmowaniu, kroki 1, 2 oraz 3 są dokładnie takie same, jak w przypadku dodawania. Jedyną różnicą jest w kroku 4, gdzie zamiast znaku plus „+” musimy wpisać znak minus „-”, na przykład:



## Mnożenie

Mnożenie przebiega tak samo jak dodawanie i odejmowanie. Należy jedynie zamienić znak działania na gwiazdkę \*, która symbolizuje mnożenie:



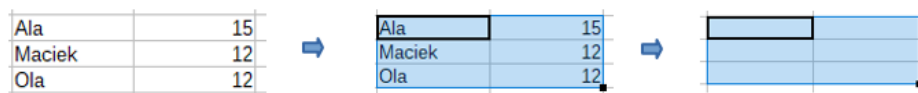
## Dzielenie

W dzieleniu wystarczy w miejscu znaku wpisać ukośnik „/” symbolizujący dzielenie:



## Usuwanie zawartości komórki

Wiesz już, w jaki sposób wypełniać komórki. Usuwanie komórek jest również bardzo proste. Należy zaznaczyć komórki, których zawartość chcesz usunąć, a następnie wcisnąć klawisz Delete. Na przykład:



Również w arkuszu kalkulacyjnym działa skrót klawiszowy **Ctrl+Z**, który umożliwia cofnięcie ostatniej wykonanej operacji.

## Edycja zawartości komórki

Edycja zawartości komórki jest bardzo przydatna, jeśli chcesz zaktualizować jej wartość lub poprawić błąd. Aby zmienić wartość komórki, można dwukrotnie kliknąć w wybraną komórkę. Pojawi się migający kursor tekstowy, co oznacza, że teraz możemy edytować zawartość komórki, tak jak w przypadku zwykłego tekstu.

Na przykład, jeśli wpisaliśmy do komórki nazwę miasta „Kra<sup>ó</sup>”, zamiast „Kra<sup>ó</sup>”, możemy łatwo poprawić ten błąd. Używając klawisza Backspace (czyt. bakspejs), usuwamy niechcianą spację.

Pamiętaj, aby wszystkie operacje na komórkach potwierdzać klawiszem Enter.

### Zadanie 1.

Przepisz następującą tabelę, rozpoczynając od kolumny A i wiersza 1:

	A	B	C
1	Dodawanie		
2	Liczba 1	Liczba 2	Wynik
3		3	7
4		3,5	2,7
5		12	32
6		457	873

Dodaj do siebie dwie liczby w każdym wierszu.

Wyniki zapisz w kolumnie Wynik.

### Zadanie 2.

Przepisz następującą tabelę poniżej poprzedniej, rozpoczynając od kolumny A i wiersza 7:

7	Odejmowanie		
8	Liczba 1	Liczba 2	Wynik
9		10	5
10		12,4	3,2
11		454	326
12		9831	3253

Odejmij liczbę 2 od liczby 1 w każdym wierszu.

Wyniki zapisz w kolumnie Wynik.

### Zadanie 3.

Przepisz następującą tabelę poniżej poprzedniej, rozpoczynając od kolumny A i wiersza 13:

13	Mnożenie		
14	Liczba 1	Liczba 2	Wynik
15	6	7	
16	12	3	
17	12,6	3,2	
18	231	43	

Pomnóż przez siebie obie liczby w każdym wierszu.

Wyniki zapisz w kolumnie Wynik.

#### Zadanie 4.

Na końcu przepisz następującą tabelę, rozpoczynając od kolumny A i wiersza 19:

19	Dzielenie		
20	Liczba 1	Liczba 2	Wynik
21	45	9	5
22	64	8	8
23	456	3	152
24	321	32	10,03125

Podziel liczbę 1 przez liczbę 2 w każdym wierszu.

Wyniki zapisz w kolumnie Wynik.

Zapisz arkusz w swoim folderze na dysku.

Plik nazwij „Arkusz2\_Imię\_Nazwisko\_Klasa”.