

INFORMATYKA

4
KLASA

EWA GAŚIENICA-SAMEK
ANDRZEJ GAŚIENICA-SAMEK
KATARZYNA OLĘDZKA

Wersja do recenzji
Warszawa, 2019-06-04

instaKod

PROGRAMOWANIE OD PODSTAW

Kontakt:

Ewa Gąsienica-Samek

tel. 666603818

egsamek@gmail.com

Spis treści

WPROWADZENIE

1	Bezpieczna praca z komputerem.	5
2	Korzystanie z Internetu.	8
3.	Na początku były dane – zera i jedyńki.	12

PODSTAWY PROGRAMOWANIA

1	InstaKod 1. Wejście i wyjście z programu.....	17
	Program, dane wejściowe i dane wyjściowe. Wprowadzenie do języka Assembly. Pudełko jako miejsce w przechowywujące liczbę. Instrukcje wejścia i wyjścia. Logowanie na konto ucznia Instakod.	
2	InstaKod 2. Wyjście z programu, stałe i zmienne.	34
3	InstaKod 3. Operacje arytmetyczne 1.	39
	Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę.	
4	InstaKod 4. Operacje arytmetyczne 2.	45
	Zwiększanie i zmniejszanie wartości pudełka o podaną liczbę. Krok dalej.	
5	InstaKod 5. Operacje arytmetyczne 3.	52
	Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka.	
6	InstaKod 6. Operacje arytmetyczne 4.	60
	Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka. Krok dalej.	
7	InstaKod 7. Suma i różnica1.	67
	Suma i różnica w pudełku pomocniczym. Przypisanie wartości do pudełka.	
8	InstaKod 8. Suma i różnica 2.	75
	Suma i różnica w pudełku pomocniczym. Krok dalej.	

9	InstaKod 9. Instrukcja warunkowa 1.	81
	Pierwsze starcie. Jeżeli warunek jest spełniony, to podejmij działanie, w przeciwnym razie zakończ program.	
10	InstaKod 10. Instrukcja warunkowa 2.	90
	Krok dalej. Jeżeli warunek jest spełniony, to podejmij działanie I, w przeciwnym razie podejmij działanie II.	

TECHNOLOGIE INFORMACYJNE

1	Edytor graficzny. Paint.....	102
	Rysowanie linii i krzywych. Rysowanie kształtów. Powielanie wzorów. Pola tekstowe. Projekt międzyprzedmiotowy.	
2	Edytor tekstowy. LibreOffice Writer / MS Word.....	116
	Dokumenty tekstowe. Polskie litery, znaki specjalne, skrótów klawiszowe. Formatowanie czcionek. Listy numerowane. Formatowanie tekstu, akapity, wcięcia, interlinie. Projekt międzyprzedmiotowy.	
3	Prezentacje. LibreOffice Impress / MS Powerpoint.....	133
	Dodawanie i usuwanie slajdu. Wstawianie tekstu. Wklejanie ilustracji. Zasady tworzenia i tryb prezentacji. Projekt grupowy międzyprzedmiotowy.	
4	Arkusz kalkulacyjny. LibreOffice Calc / MS Excel.	145
	Komórki, kolumny, wiersze i ich formatowanie. Liczby w komórkach, operacje dodawania, odejmowania, dzielenia i mnożenia wartości dwóch komórek.	

WPROWADZENIE

Witaj na lekcjach informatyki!

Czym będziemy 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, ale niesie też ze sobą wiele niebezpieczeństw.



Większą część roku szkolnego poświęcimy na poznawanie podstaw programowania. Informatyka, podobnie jak matematyka, jest nauką ścisłą. Co więcej, na naszych 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 wypadku 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**

Platforma InstaKod, daje Ci dostęp do przykładów, zadań i środowiska programistycznego. w którym będziesz je rozwiązywać. Na platformie będziesz zaliczać sprawdziany i quizy. W zakładce „Oceny” znajdziesz informacje o liczbie zaliczonych zadań, terminach prac domowych oraz proponowanych przez system ocenach.



Zeszyt ćwiczeń zawiera karty pracy do rozwiązania bez dostępu do komputera.

Zeszyt do informatyki okaże się nieocenioną pomocą przy rozwiązywaniu zadań programistycznych. Szkic rozwiązania każdego zadania powinien najpierw 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.

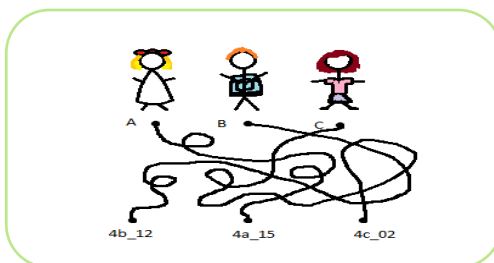
Naukę informatyki rozpoczniemy od zapoznania się z zasadami pracy z komputerem. Przeczytaj uważnie regulamin pracowni. Następnie przyjrzyj się swojemu komputerowi i zastanów się czy umiesz nazwać wszystkie jego elementy.

Jak zachować się w pracowni?

1. Stosuj się do poleceń nauczyciela. Pracuj tak, by nie przeszkadzać innym. Pamiętaj, to jest miejsce wyętej 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.

Połącz każdego ucznia z jego nazwą użytkownika.



Bezpieczne hasło

Hasła służą zabezpieczeniu naszych 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, bo wtedy łatwo je odgadnąć. Przyjmuje się, że hasło powinno mieć co najmniej 8 znaków. Nie powinno zawierać liter tworzących łatwe do odgadnięcia wyrazy. Do jego tworzenia nie używaj swojego imienia, nazwiska ani pseudonimu. Dobrze jest mieć różne hasła dla oddzielnych systemów. Nikomu nie podawaj swojego hasła.

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

Zadanie 2. Przy każdym hasle zapisz, czy można je uznać za bezpieczne, czy nie. Weź pod uwagę opisane wyżej zasady.

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

Zapamiętanie hasła

Hasło powinno być trudne do odgadnięcia przez innych, ale jednocześnie łatwe do zapamiętania przez Ciebie. 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. → kwtjjwjp

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 cytaty dla każdego z przykładów.
Zastosuj po kolei wszystkie trzy zasady i wygeneruj hasło.
Pamiętaj, nie podawaj swojego prawdziwego hasła.

Polecenie	Przykład 1	Przykład 2	Przykład 3
Napisz powiedzenie lub cytaty.			
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 rozwiniesz swoje umiejętności techniczne o pisanie tekstów, przygotowywanie rysunków, a przede wszystkim o podstawy programowania. Wkraczysz w świat informatyki.

Zadanie 4. Uzupełnij tekst odpowiednimi ikonkami: , , , .

Na lekcjach dowiesz się jak pisać,
przygotowywać, a przede wszystkim nauczysz się
programować. Wkraczasz w informatyki.

2

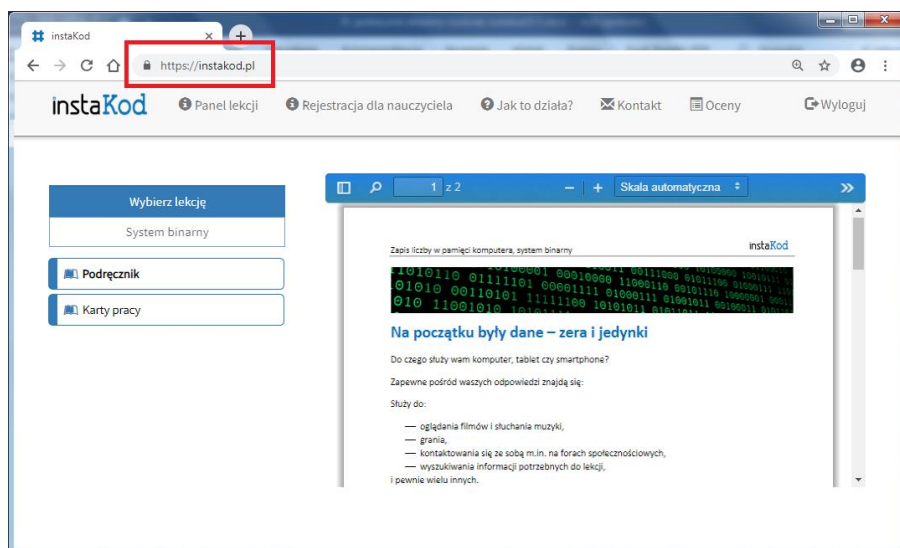
Korzystanie z Internetu

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: 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 dobre. Naszym obowiązkiem jest nauczenie się, w jaki sposób można unikać niebezpiecznych treści.

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 podaje listę adresów stron pasujących do zapytania.

Poniżej znajduje się lista stron, z których będziemy korzystać:

- instakod.pl – to platforma, na której będziesz rozwiązywać zadania z informatyki. Znajdują się na niej materiały przygotowane specjalnie dla Ciebie;
- 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, za przykład można podać Wikipedię – bardzo popularną encyklopedię internetową. Być może któryś z artykułów dotyczy interesującego Cię tematu.

Zadanie 1. Adresy stron internetowych.

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

1. <http://sieciaki.pl> – strona dla dzieci o bezpieczeństwie w sieci
2.
3.
4.
5.
6.

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. Napraw błędy.

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 zwięzłe.
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.

1	2	3	4

3

Na początku były dane – zera i jedynki

Zastanów się, do czego służy Ci komputer, tablet czy smartphone.

Zapewne udzieliłeś 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 informacji. Miejsce, w którym przechowywane są dane, zwane jest pamięcią.



Zastanówmy się nad poniższymi pytaniami.

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

Na pierwsze pytanie odpowiemy już dziś. Na drugie zaś, odpowiedź będziemy budować 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 nazywamy 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
na twoim
telefonie

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
twardego dysku
na komputerze

4 TB, czyli
16 terabajtów

to jest około
4 000 000 000 000
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.

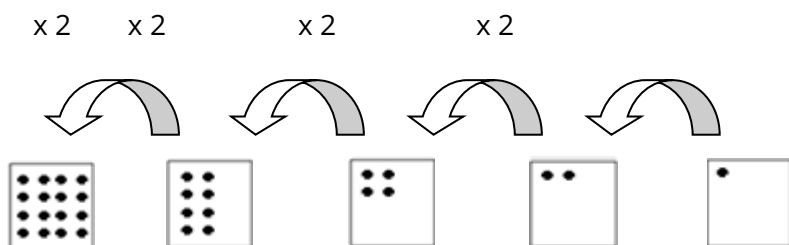
$$12 = 1 \times 10 + 2 \times 1$$

Zapis ten możemy przełożyć na gotówkę, którą możemy trzymać wyłącznie w monetach 1 zł, oraz banknotach 10 zł, 100 zł, 1000 zł, itd.

12 zł to oczywiście 1 banknot 10 zł i 2 monety 1 zł.

Ale jak zapisać te 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 wybraliśmy, 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><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. Wiemy, 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 ($28 - 16 = 12$).

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

Pozostają jeszcze 4 kropki ($12 - 8 = 4$). Bierzemy więc kartę **4**.

28 reprezentujemy kartą 16, kartą 8 oraz kartą 4.

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

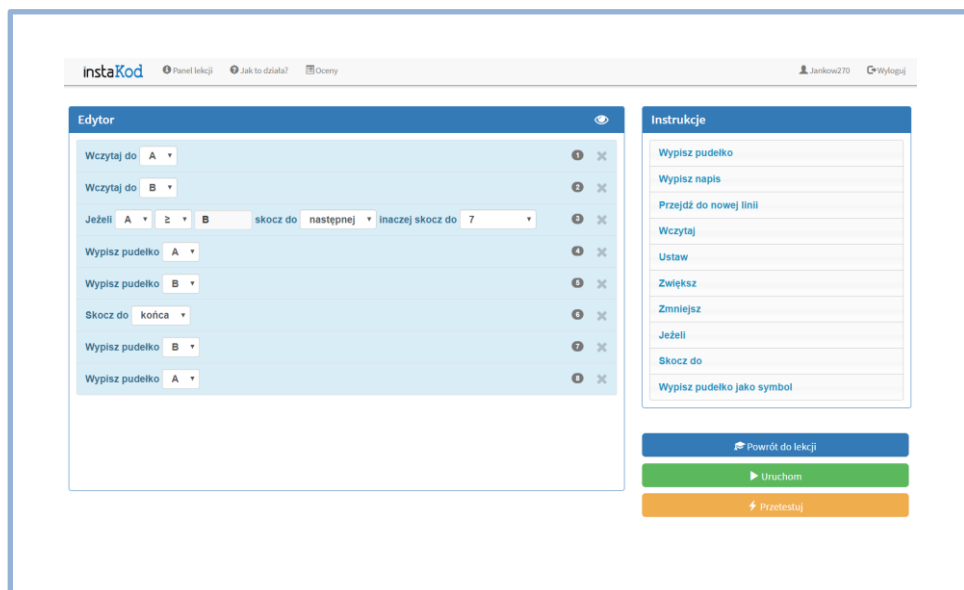
Jej zapis w systemie binarnym wygląda więc tak:

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



Przejdź do **zeszytu ćwiczeń** i wypełnij karty pracy.

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 na Twoim koncie ucznia na instakod.pl. Sprawdzaj na bieżąco swoje oceny w systemie. Pomocne w budowaniu rozumienia podstaw programowania będą karty pracy, które znajdziesz w zeszytach ćwiczeń.

Do dzieła!

InstaKod 1

Wejście i wyjście z programu

W poprzednim rozdziale dowiedziałeś się, jak zapisane są dane liczbowe w pamięci komputera. Czas by 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.

By bezpiecznie i szybko zawieźć pasażerów na miejsce, kierowca musi znać **trasę**. W przeciwnym razie, zamiast w góry, zawiezie swoich pasażerów nad morze albo wjedzie na drogę gruntową i zakopie się w błocie. Po drugie, musi 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, skrócić 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.

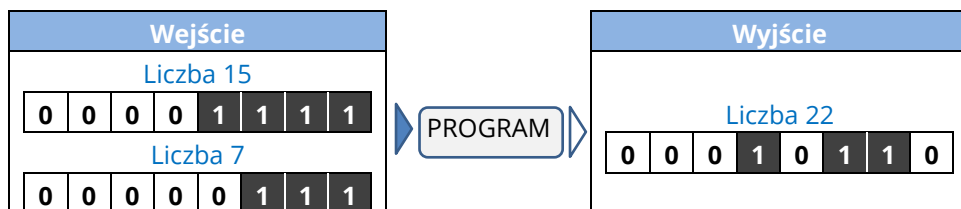
Wejście do programu

Wejście to dane podane przez użytkownika do programu.

Wyjście z programu

Wyjście to dane wypisane przez program.

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 od nas 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.



Logowanie na konto uczniowskie

Platforma, na której będziesz uczyć się programowania, nazywa się **InstaKod**.

Twoje konto zostanie założone przez nauczyciela, który przekaze Ci login oraz hasło.

Janek, uczeń klasy IV otrzymał swój login i hasło w formie wydruku z systemu. Otrzymałą kartkę chłopiec wkleił do zeszytu od informatyki.

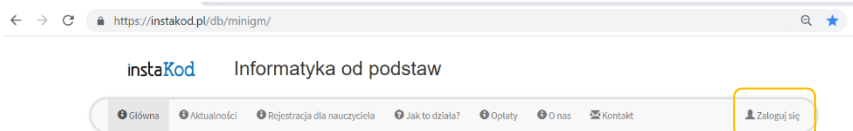


Aby zalogować się na swoje konto, postępuj zgodnie z następującymi krokami:

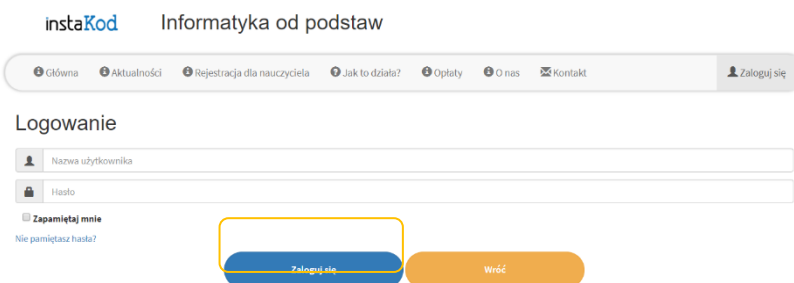
1. Otwórz przeglądarkę (np. Edge) i wejdź na stronę www.instakod.pl.

Wiesz już, jaka jest różnica między wpisaniem nazwy strony w oknie przeglądarki i wyszukiwarki. Upewnij się, że wpisujesz ją w oknie przeglądarki, by nie szukać jej wśród stron, które zawierają podobne frazy do „instakod”.

2. Wciśnij guzik „Zaloguj się”.



3. Wpisz swój login w oknie „Nazwa użytkownika” i wypełnij pole „Hasło”. Następnie wciśnij przycisk „Zaloguj się”.



Gratulacje, udało Ci się zalogować na swoje konto!

Pamiętaj:

- Jeśli zdarzy Ci się zapomnieć login lub hasło, zgłoś to nauczycielowi.
- Nie udostępniaj nikomu swojego loginu i hasła.
- Jeśli podejrzewasz, że ktoś ma dostęp do Twojego konta, poproś nauczyciela o nowe hasło.

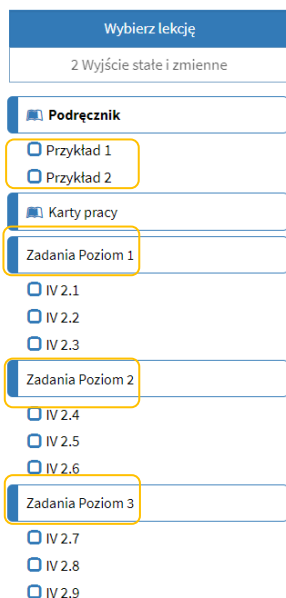
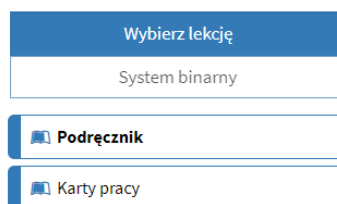
Panel ucznia

Menu **Wybierz lekcję**, służy do przechodzenia między rozdziałami. Wybierz rozdział, który aktualnie realizujecie w klasie.

W **Podręczniku** znajdziesz omówienie zagadnień związanych z danym rozdziałem.

Karty pracy to ćwiczenia, które należy wydrukować i wykonać na papierze, bez wsparcia komputera.

Rozdziały z podstaw programowania zawierają zadania programistyczne podzielone na:



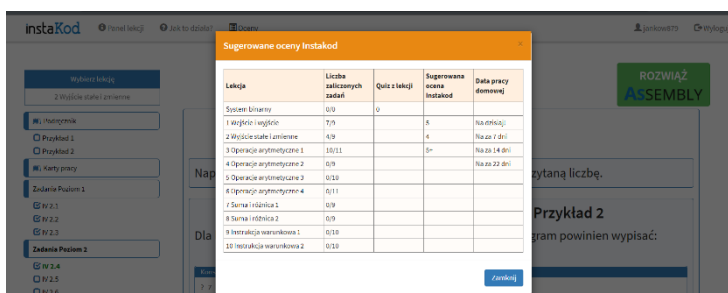
Przykłady będą omawiane przez nauczyciela. Wzorcowe rozwiązania znajdziesz w podręczniku.

Zadania poziomu 1
to zadania najprostsze.

Zadania poziomu 2
to zadania średniej trudności.

Zadania poziomu 3
to zadania trudne.

System proponuje oceny za Twoją pracę. Możesz je sprawdzić w zakładce [Oceny](#).



Lekcja	Liczba zaliczonych zadań	Quiz z lekcji	Sugerowana ocena InstaKod	Data pracy domowej
System Unitary	0/0	0		
1 Wzrost i waga	7/5		5	Nu dntuaj
2 Wzrost stały i zmienne	4/9		4	Nu za 7 dni
3 Operacje arytmetyczne 1	10/11		5+	Nu za 14 dni
4 Operacje arytmetyczne 2	0/9			Nu za 22 dni
5 Operacje arytmetyczne 3	0/10			
6 Operacje arytmetyczne 4	0/11			
7 Seria i różnica 1	0/9			
8 Seria i różnica 2	0/9			
9 Interakcja warunkowa 1	0/10			
10 Interakcja warunkowa 2	0/10			

Za rozwiązane i zaliczone przez InstaKod zadania otrzymasz sugerowaną ocenę:

- 2 za 2 zadania,
- 3 za 3 zadania,
- 4 za od 4 do 6 zadań,
- 5 za od 7 do 9 zadań,
- 5+ za 10 lub więcej zadań.

Wyjście stałe, czyli komunikaty z programu

Instrukcja „Wypisz napis”

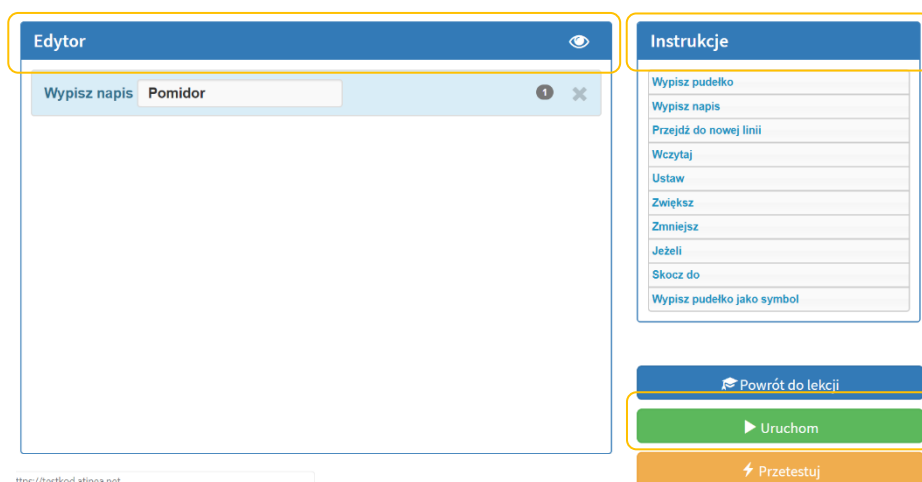
Naukę programowania rozpoczniemy od tworzenia prostego programu wczytującego liczby i wypisującego liczby oraz słowa.

Twoim pomocnikiem w rozmowie z komputerem będzie język wizualny Assembly. Jest to język zrozumiały dla komputera.

Po zalogowaniu się na swoje konto na platformie instakod.pl w menu [Wybierz lekcję](#) zaznacz rozdział 1 „Wejście i wyjście”, który aktualnie realizujecie w klasie. Po lewej stronie znajdują się zadania programistyczne, które są podzielone na przykłady i zadania trzech poziomów trudności.

Po kliknięciu na zadanie, pojawia się jego treść. W prawym górnym rogu ekranu znajduje się duży, zielony przycisk „Rozwiąż Assembly” służący do przejścia do edytora. Klikając na przycisk zostaniesz przeniesiony do widoku rozwiązywania zadania. Z lewej strony ekranu pojawi się duża ramka „Edytor”, a po prawej stronie mniejsza ramka „Instrukcje”. W polu „Edytor” będziesz tworzyć swoje programy, przeciągając instrukcje z mniejszej ramki.

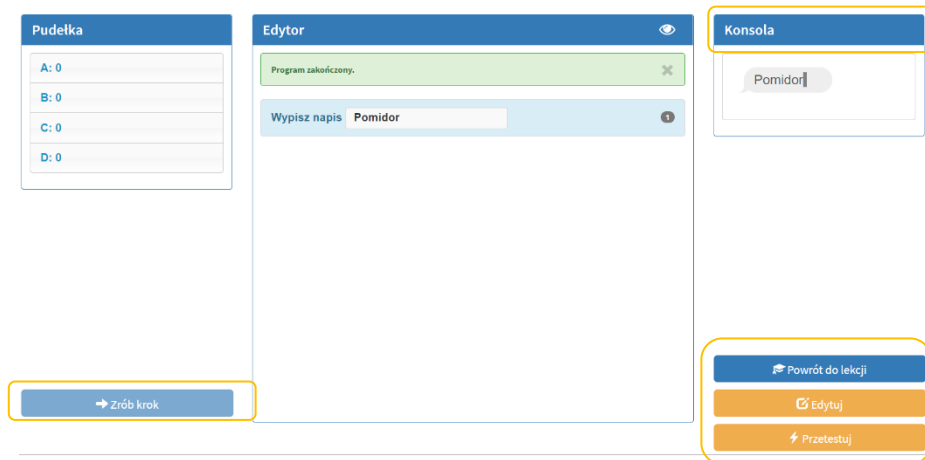
Napiszemy teraz nasz pierwszy program. Przeciągnij do edytora instrukcję [Wypisz napis](#). Instrukcja ta służy do wypisywania na ekran tekstu z białego pola. Tekst wpisujemy z klawiatury, która pojawia się na ekranie lub bezpośrednio z klawiatury komputera. Wpisz słowo *Pomidor*.



Gratulacje! Zbudowałeś właśnie swój 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](#).



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

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.

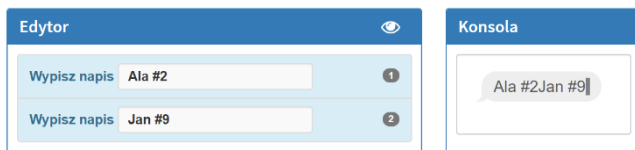
Instrukcja „Przejdź do nowej linii”

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

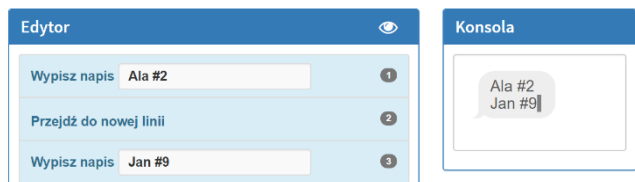
Ala #2

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”.



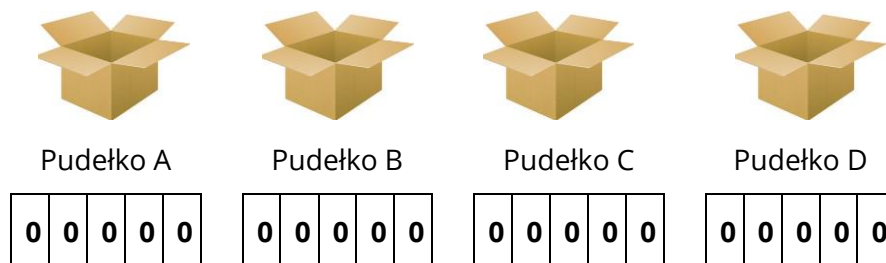
Jeśli program zawsze wypisuje to samo, niezależnie od tego jakie dostanie dane wejściowe (pytanie), mówimy, że jest to **wyjście stałe**. Jest to trochę jak gra w pomidora. W tej grze, na każde pytanie zawsze odpowiadamy *Pomidor*. Komputer nigdy nie przegrałby w takiej grze. Wystarczyłoby, zbudować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

Pudełko. Wejście. Instrukcja Wczytaj

Wiesz już, jak wypisywać komunikaty takie, jak na przykład *Jan, 786, #Kasia* czy *35a*.

Dowiesz się teraz, jak przetwarzać dane, czyli pisać programy, które przyjmą dane wejściowe, coś z nimi zrobią i zwrócą dane wyjściowe. Dane wejściowe zapisujemy w pudełkach. Pudełko to miejsce w pamięci komputera, w którym przechowywana jest liczba.

W Assembly mamy do dyspozycji 4 pudełka. W każdym z nich możemy 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?

Pudełko	
A:	0
B:	0
C:	0
D:	0

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

Instrukcja „Wczytaj” zapisze w wybranym pudełku liczbę pobraną od użytkownika. Do wyboru mamy pudełko: A, B, C i D.

Wczytaj do	A ▼
	A
	B
	C
	D

Zmienna, inaczej pudełko

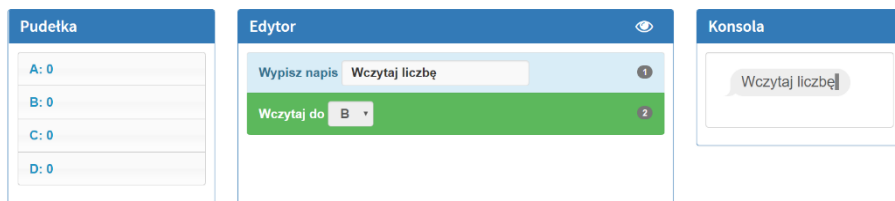
Pudełko to miejsce w pamięci komputera, w którym przechowywana jest liczba. Każde pudełko ma swoją nazwę oraz rozmiar (maksymalną wielkość związaną z liczbą komórek pamięci przypisaną do pudełka). W Assembly dostępne są pudełka: A, B, C i D, z których każde ma rozmiar 16 komórek pamięci (bitów) oraz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

Zadanie

Napisz program, który wyświetli komunikat proszący użytkownika o wczytanie liczby i zapisze ją w pudełku B.

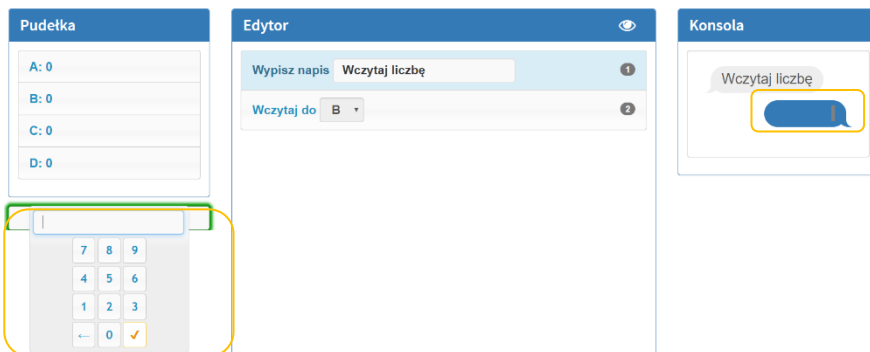
Do wyświetlania komunikatów do użytkownika służy instrukcja „Wypisz napis”. Aby pobrać liczbę od użytkownika i zapisać ją w pudełku B, należy użyć instrukcji `Wczytaj do B`.

Uruchommy poniższy program i wykonajmy pierwszy krok. Na konsoli pojawi się szary dymek „rozmowy” komputera z napisem *Wczytaj liczbę*.



Zróbmy teraz kolejny krok. Na konsoli pojawi się niebieski dymek rozmowy użytkownika. W ten sposób komputer prosi użytkownika o podanie liczby. Równoległe, pod ramką `Pudełka`, pojawi się klawiatura, na której możemy wpisywać liczby.

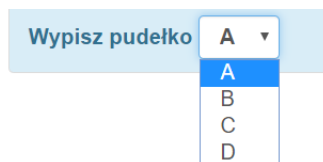
Wyberzmy liczbę 8.



Po wczytaniu liczby zaktualizowane zostaną wartości pudełek. W ramce „Pudełka” obok pudełka B, znajduje 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 pokazać 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.



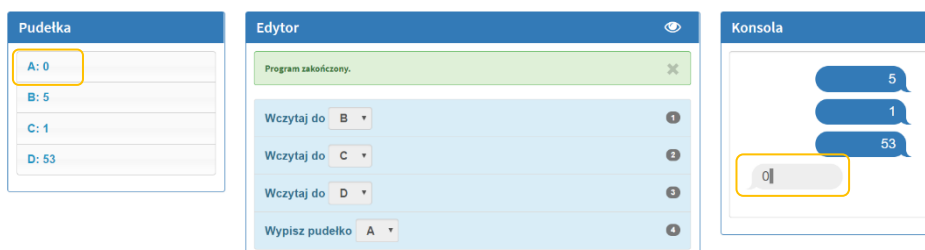
Zadanie

Napisz program, który wczyta liczby do pudełek: B, C i D i 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. W przypadku powyższego programu jest to liczba 5.

Wyjście z 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 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.

Rozwiązania przykładów

Jak powstają systemy informatyczne?

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

Zwykle zaczyna się od opracowania założeń. Tworzy się opis, jak system ma działać, a następnie „rozwiązanie”, czyli plan tego, jak system ma być skonstruowany. Następnie przechodzi się do implementacji (pisania kodu programu). Już w trakcie pisania kodu testuje się system, 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, tak jak duże systemy informatyczne 😊.

Schemat rozwiązania zadania

- Przeczytanie ze zrozumieniem treści zadania,
- sformułowanie konkretnego wejścia i oczekiwanego wyjścia,
- wymyślenie rozwiązania,
- zapisanie rozwiązania w formie diagramu w zeszycie,
- napisanie kodu (implementacja),
- uruchomienie programu dla założonego wejścia i porównanie wyjścia z oczekiwanym,
- automatyczne przetestowanie,
- przeanalizowanie testów i błędów,
- korekta,
- dalsze testowanie i korekta, aż do skutku.

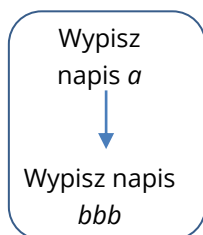
Przykład 1.

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
	Komputer	Użytkownik
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> .	a	
	bbb	

Diagram rozwiązania i jego implementacja.



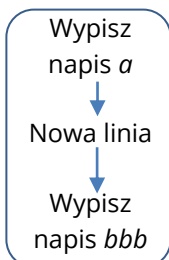
Zwróć uwagę, że przy każdym uruchomieniu programu na konsoli ma się pojawić ten sam tekst. Mówimy, że jest to wyjście stałe. Do wypisywania tekstu służy instrukcja „Wypisz napis”.

```
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 liniijkach.



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.

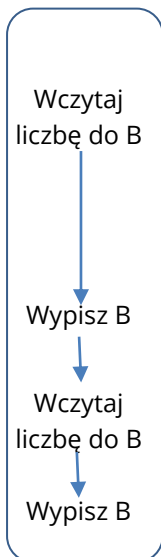
Dzielimy zadanie na kroki, które kolejno musimy wykonać.

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.

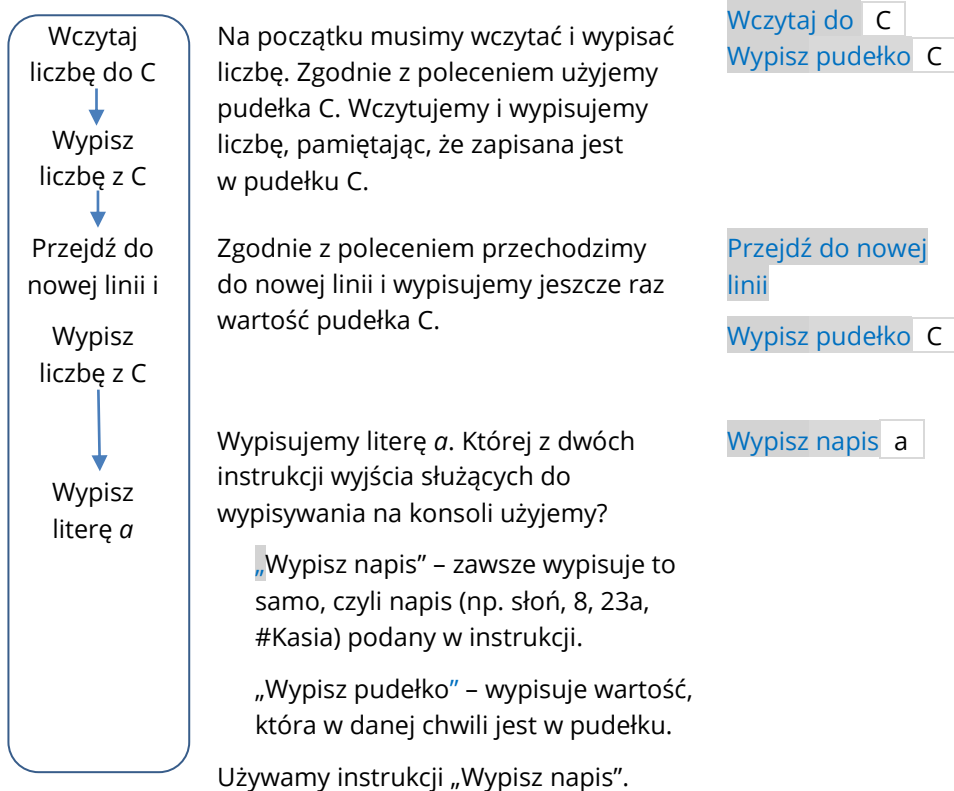
Dzielimy zadanie na kroki, które kolejno musimy wykonać.

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.	

Diagram rozwiązania i jego implementacja.





instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

Zadania InstaKod 1. Wejście i wyjście z 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 literę *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, wypisze literę *c*, wartość pudełka A oraz napis *xxx*.

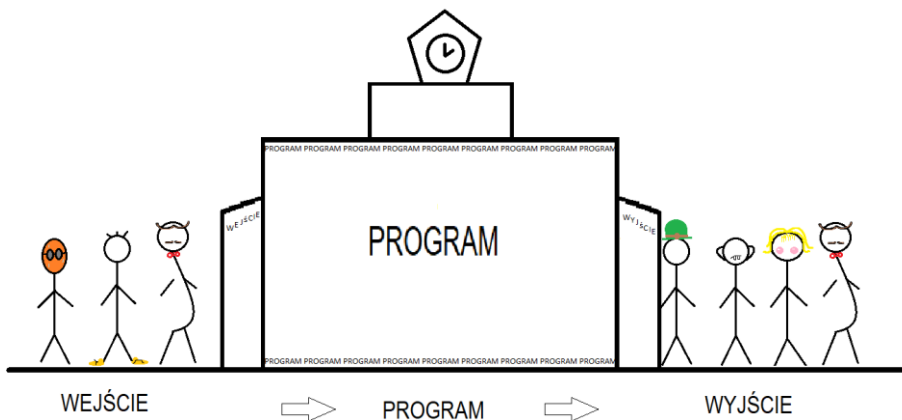
IV 1.8 Napisz program, który wczyta liczbę do pudełka B i wypisze jego wartość. Następnie ponownie wczyta liczbę do pudełka B i wyświetli jego nową wartość.

IV 1.9 Napisz program, który wczyta liczbę do pudełka A i wypisze literę *x*. Następnie przejdzie do nowej linii i dwa razy wypisze wczytaną liczbę.

InstaKod 2

Wyjście z programu, stałe i zmienne

Rolą kierowcy autobusu jest przewiezienie pasażerów z punktu A do punktu B. Podobnie rolą programisty jest wygenerowanie oczekiwanych danych wyjściowych z podanych przez użytkownika danych wejściowych.



W poprzednim rozdziale poznałeś podstawowe instrukcje w Assembly, które pozwolą Ci wczytywać i wypisywać dane oraz komunikować się z użytkownikiem:

Wejście do programu

[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 skasowana.

Wyjście z 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 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 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.

Pamiętaj, do którego pudełka wczytałeś liczbę. Jeśli wczytałeś liczbę do pudełka C i chcesz ją wypisać, to używając instrukcji „Wypisz pudełko”, musisz zaznaczyć pudełko 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 pierwsza zostanie 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 powinien wypisać na wyjściu ten sam komunikat, to jest liczbę 2.	2

Przy pisaniu każdego 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?

Jeśli odpowiedź na pierwsze pytanie brzmi tak, używamy instrukcji

`Wypisz napis` .

Jeśli odpowiedź na drugie pytanie brzmi tak, używamy instrukcji

`Wypisz pudełko` .

W przypadku naszego przykładu, za każdym razem chcemy wypisać to samo. Używamy 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.

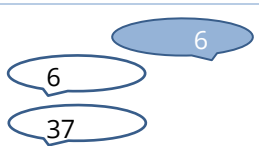
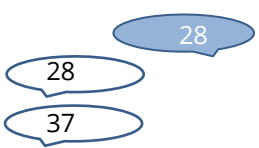
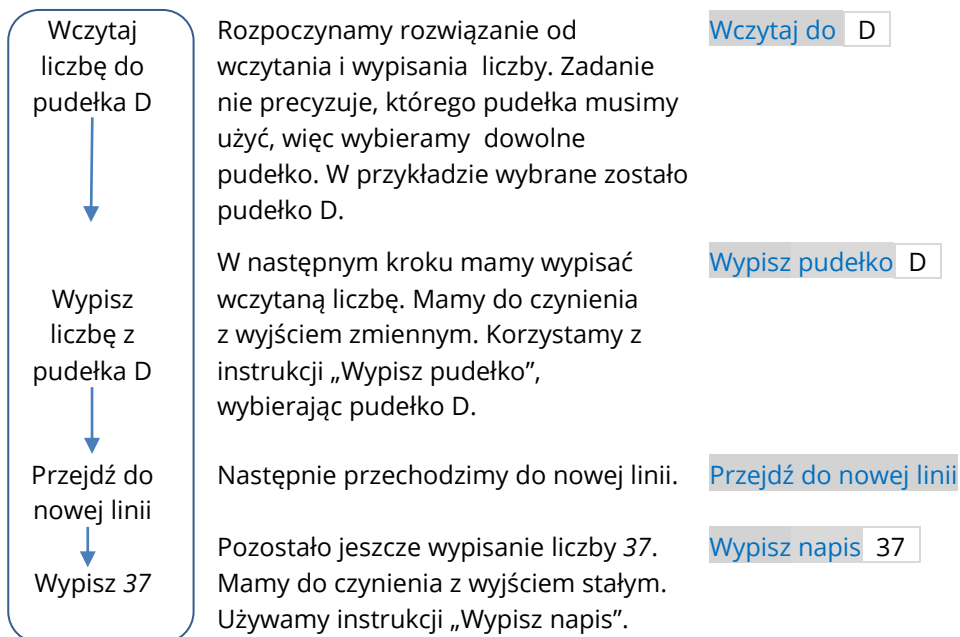
Wejście i wyjście	Konsola
Dla podanej na wejściu liczby 6 program powinien w osobnych wierszach wypisać na wyjściu 6 oraz 37.	
Dla podanej na wejściu liczby 28 program powinien w osobnych wierszach wypisać na wyjściu 28 oraz 37.	

Diagram rozwiązania i jego implementacja.





instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

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 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 i trzykrotnie ją wypisze.
- 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 wypisze na konsoli wczytaną liczbę, liczbę 7 oraz ponownie wczytaną liczbę.

Poziom 3

- IV 2.7 Napisz program, który wczyta liczbę i wypisze liczbę 3, wczytaną liczbę oraz liczbę 3.
- IV 2.8 Napisz program, który wczyta liczbę i wypisze **x**, wczytaną liczbę oraz liczbę 12. Następnie przejdzie do nowej linii i ponownie wypisze wczytaną liczbę.
- IV 2.9 Napisz program, który wczyta liczbę, wypisze napis **ab3**, wczytaną liczbę i w nowej linii liczbę 6.

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 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 jest **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 **stałe 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”.

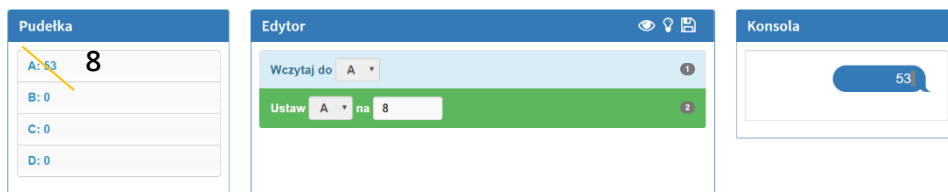
Ustaw A na 8 zapisuje w pudełku A liczbę 8.
Liczba, która wcześniej znajdowała się w pudełku A, zostaje usunięta.
Mówimy, że przypisaliśmy liczbę 8 do pudełka A.

Czym różnią się instrukcje „Ustaw A na 8” i „Zwiększ A o 8”?

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

Instrukcja **Zwiększ** A o 8 sprawdzi, jaka liczba jest w pudełku A i doda do niej 8, zapisując wynik w pudełku A.

W przypadku, gdy w pudełku A jest liczba 0, to po wykonaniu obu instrukcji, znajdzie się tam liczba 8. Jeśli w A jest liczba różna od 0, np. 53, to po wykonaniu instrukcji **Ustaw** A na 8, liczba 53 zamieni się na 8.



Po wykonaniu instrukcji **Zwiększ** A o 8 liczba 53 zamieni się na 61.



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 (bo $3 + 6 = 9$).	

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

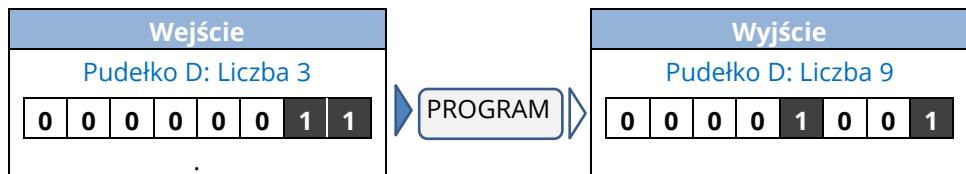
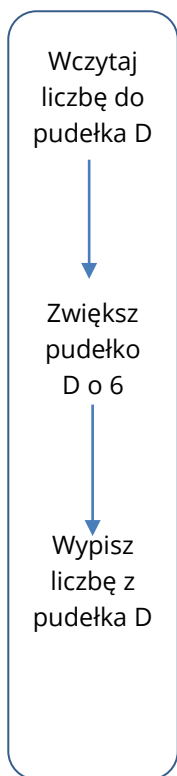


Diagram rozwiązania i jego implementacja.



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

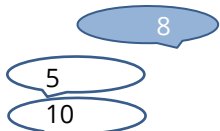
Przykład 2.

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 ($8 - 3 = 5$) oraz 10 ($5 + 5 = 10$).	

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

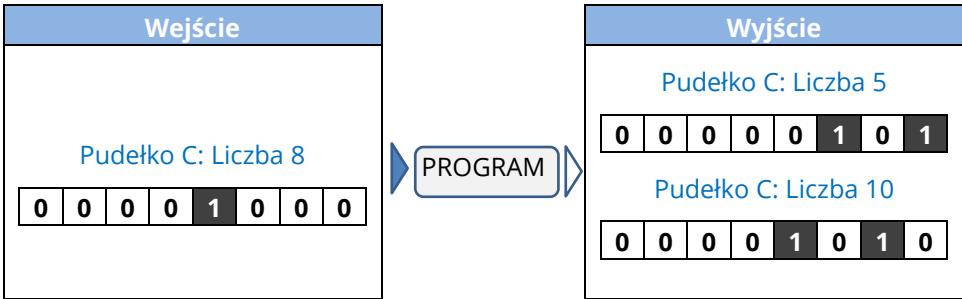
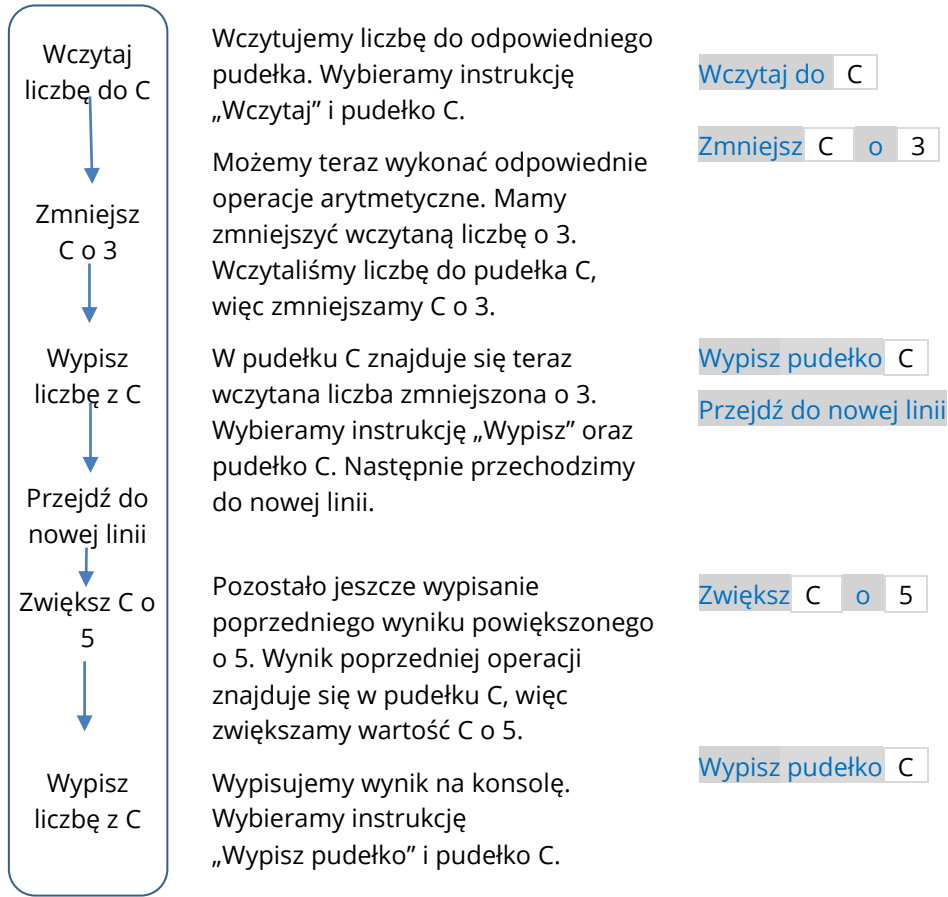


Diagram rozwiązania i jego implementacja.





instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

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

Poziom 2

- IV 3.4 Napisz program, który wczyta liczbę do pudełka C i wypisze wartość pudełka C. Następnie zwiększy ją o 10 i 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 wypisze wynik oraz liczbę 5.

Poziom 3

- IV 3.7 Napisz program, który wczyta liczbę do pudełka B i wypisze pudełko B. Następnie zwiększy B o 8, zmniejszy o 2 i wypisze końcową wartość pudełka B.
- IV 3.8 Napisz program, który wczyta liczbę do pudełka C i zmniejszy pudełko C o 2. Następnie wypisze dwukrotnie nową wartość pudełka C oraz liczbę 6.
- IV 3.9 Napisz program, który wczyta liczbę do pudełka C i wypisze 2. Następnie w nowej linii wypisze wartość pudełka C pomniejszoną o 2.
- 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* Napisz program, który wczyta liczbę do pudełka C, wypisze literę *a* oraz wartość pudełka C. Następnie zmniejszy C o 2 i w nowej linii wypisze końcową wartość pudełka C.

InstaKod 4

Operacje arytmetyczne 2

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

Dotychczas poznaliśmy następujące pojęcia i instrukcje:

Wejście do 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 z 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 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 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 znajdującą się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

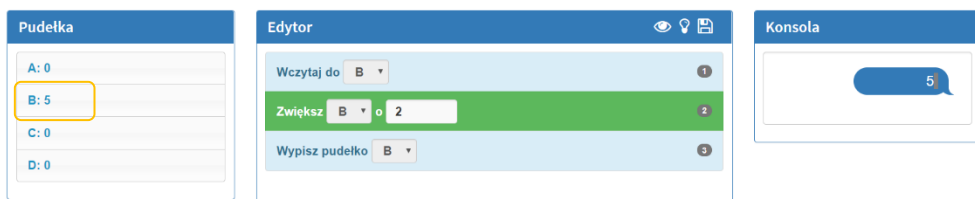
Pisząc programy, które wykonują operacje arytmetyczne na wczytanych liczbach, pamiętaj, do którego pudełka wczytałeś liczbę.

Zadanie.

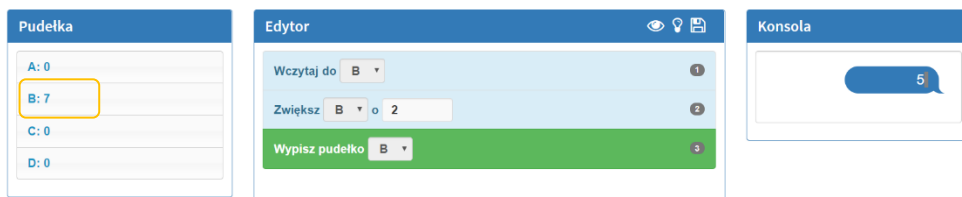
Napisz program, który wczyta liczbę i wypisze ją zwiększoną o 2.

Na początku musimy zdecydować się na pudełko, do którego wczytamy liczbę. W przykładzie wybrane zostało pudełko B. Następnym krokiem jest zwiększenie pudełka o 2, do którego wczytaliśmy liczbę. Na koniec wypisujemy nową wartość pudełka B.

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



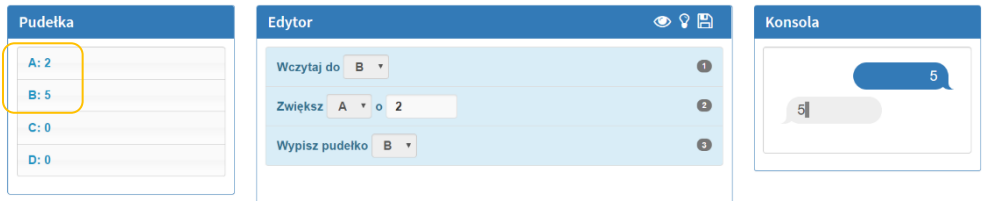
Zróbmy drugi krok. Liczba w pudełku B zostanie zwiększona o 2, przyjmie wartość 7.



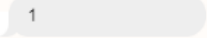
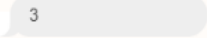
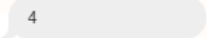
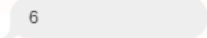
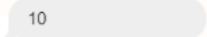
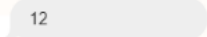
W trzecim kroku na konsoli zostanie wypisana wartość pudełka B.



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 to jest tam 0) zostanie powiększona o 2.



Takie rozwiązanie nie przejdzie testów.
Po próbie przetestowania otrzymamy poniższy komunikat.

Testowanie programu		
Twój program nie przeszedł poniższych testów		
Rezultat	Twoja odpowiedź	Prawidłowa odpowiedź
Zła odpowiedź	 	 
Zła odpowiedź	 	 
Zła odpowiedź	 	 

Dla wczytanej liczby 1 oczekiwany wynik to 3 ($2+1=3$), a nasz wynik to 1.
Dla wczytanej liczby 4 oczekiwany wynik to 6 ($2+4=6$), a nasz wynik to 4.
Dla wczytanej liczby 10 oczekiwany wynik to 12 ($2+10=12$), a nasz wynik to 10.

Przykład 1.

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

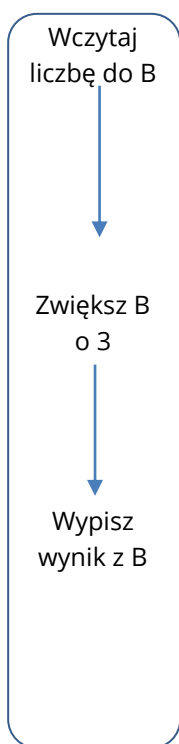
Dzielimy zadanie na poszczególne kroki:

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 ($5 + 3 = 8$).	

Diagram rozwiązania i jego implementacja.



Na początku musimy wczytać liczbę. Użyjemy do tego instrukcji „Wczytaj”. Zadanie nie precyzuje pudełka, do którego mamy wczytać liczbę. W przykładzie wybrane zostało pudełko B.

Wczytaj do B

Możemy teraz przejść do obliczeń arytmetycznych. Mamy zwiększyć wczytaną liczbę o 3. Wybieramy instrukcję „Zwiększ” oraz pudełko B i liczbę 3. Pomyśl, co by się stało, gdybyśmy zamiast pudełka B, zwiększyli pudełko A.

Zwiększ B o 3

Suma liczby wczytanej do pudełka B oraz liczby 3 znajduje się w pudełku B. Wypisujemy wynik na konsolę. Wybieramy instrukcję „Wypisz pudełko” oraz odpowiednie pudełko. Czy wiesz, dlaczego użyliśmy teraz pudełka B?

Wypisz pudełko B

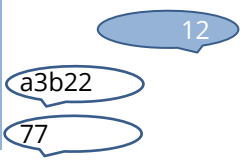
Przykład 2.

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

Dzielimy zadanie na poszczególne kroki:

Wczytaj liczbę | odejmij od niej 10 | zwiększ ją o 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 ($25 - 10 + 20 = 35$) i w nowej linii 77.	 25 a3b35 77
Dla podanej na wejściu liczby 12 program powinien wypisać na wyjściu <i>a3b</i> , liczbę 22 (bo $12 - 10 + 20 = 22$) i w nowej linii 77.	 12 a3b22 77

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

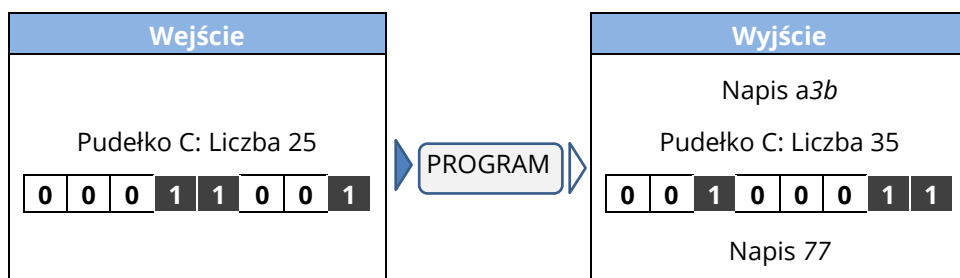
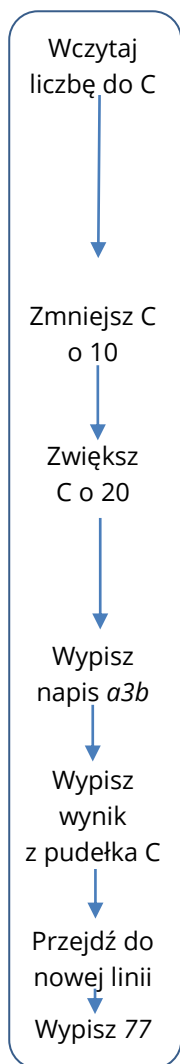


Diagram rozwiązania i jego implementacja.



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

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

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

Pozostaje nam przejść do nowej linii i wypisać 77. Do wypisywania stale tego samego komunikatu służy instrukcja wyjścia „Wypisz napis”.

Wczytaj do C

Zmniejsz C o 10
Zwiększ C o 20

Wypisz napis a3b

Wypisz pudełko C

Przejdź do nowej linii

Wypisz napis 77



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

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 wypisze wynik.
- IV 4.3 Napisz program, który wczyta liczbę, wypisze 3 oraz wczytaną liczbę pomniejszoną o 2.

Poziom 2

- IV 4.4 Napisz program, który wczyta liczbę, zmniejszy ją o 5, następnie 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 i wypisze literę a. Następnie zwiększy wczytaną liczbę o 6, przejdzie do nowej linii i wypisze wynik.
- IV 4.8 Napisz program, który wczyta liczbę, zwiększy ją o 5 i wypisze wynik. Następnie od uzyskanego wyniku odejmie 3 i wypisze końcową wartość.
- IV 4.9 Napisz program, który wczyta liczbę, zwiększy ją o 1, zmniejszy o 2 i wypisze wynik. Następnie w nowej linii wypisze liczbę 8.

InstaKod 5

Operacje arytmetyczne 3

Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka.

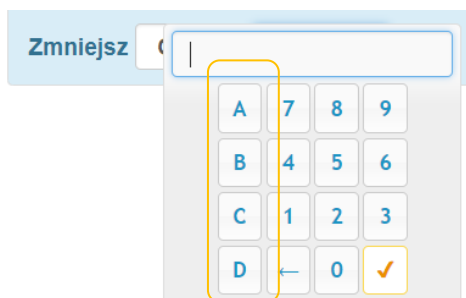
Wiesz już, jak zwiększyć lub zmniejszyć liczbę zapisaną w pudełku o wybraną liczbę.

Operacje arytmetyczne

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

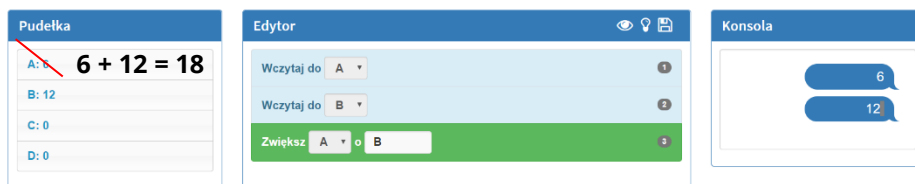
Zmniejsz C o 2 oblicza różnicę liczby znajdującej się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

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.

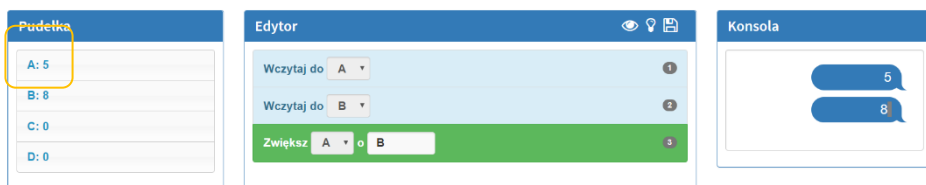
Wykonajmy instrukcję **Zwiększ** A o B. Instrukcja ta zwiększy liczbę aktualnie znajdującą się w pudełku A o wartość pudełka B.



Zadanie.

Napisz program, który policzy sumę dwóch wczytanych liczb.

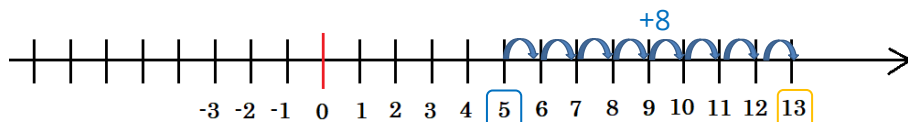
Wykorzystajmy poniższy program. Uruchommy go wczytując liczby 5 i 8. Pierwsza liczba trafi do pudełka A, a druga do pudełka B.



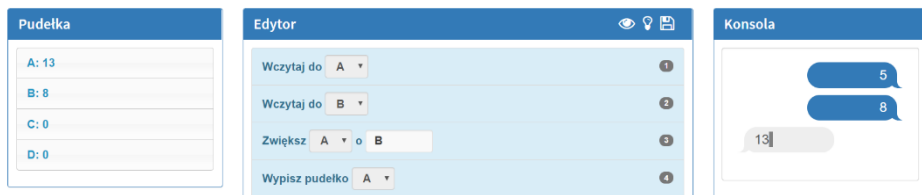
Aby policzyć sumę dwóch wczytanych liczb, musimy zwiększyć jedną z nich o drugą. Możemy zrobić to na dwa sposoby.

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

Zilustrujmy 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.

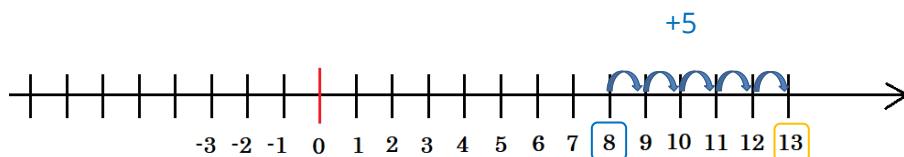


Poniższy program zapisuje sumę liczb z A i B w pudełku A. Po uruchomieniu programu i wykonaniu instrukcji **Zwiększ A o B** liczba w pudełku A zostanie zaktualizowana do 13 (bo $5 + 8 = 13$). Liczba w pudełku B nie ulegnie zmianie.

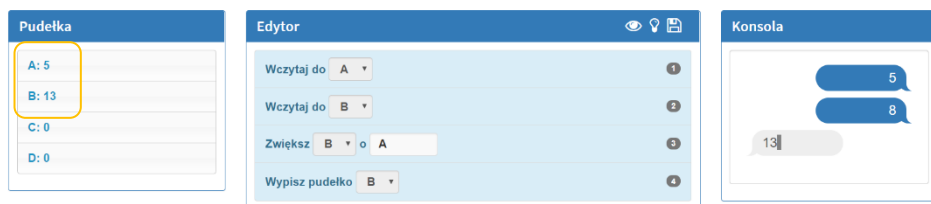


Zilustrujmy dodawanie liczb w pudełkach A i B za pomocą instrukcji **Zwiększ A o B** na osi liczbowej dla wczytanych liczb $A = 5$ i $B = 8$.

Zilustrujmy 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 uruchomieniu poniższego programu i wykonaniu instrukcji **Zwiększ B o A** liczba w pudełku B zostanie zaktualizowana do 13 (bo $5 + 8 = 13$). Tym razem liczba w pudełku A nie ulegnie zmianie.



Dotychczas poznane instrukcje uzupełniamy o instrukcję pozwalającą na zwiększenie wartości pudełka o wartość innego pudełka.

Operacje arytmetyczne

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

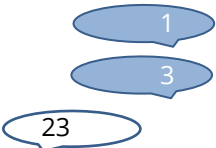
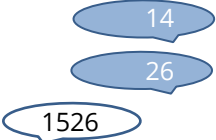
Zmniejsz C o 2 oblicza różnicę liczby znajdującej się w pudełku C oraz liczby 2 i zapisuje wynik w pudełku C.

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.

Przykład 1.

Napisz program, który wczyta liczby do pudełek A i B. Następnie 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 (bo $1 + 1 = 2$) oraz 3.	
Dla podanych na wejściu liczb 14 i 26 program powinien na wyjściu wypisać 15 (bo $14 + 1 = 15$) oraz 26.	

W jaki sposób zapisane są dane wejściowe i dane wyjściowe?

Oczywiście za pomocą bitów w komórkach pamięci poszczególnych pudełek.

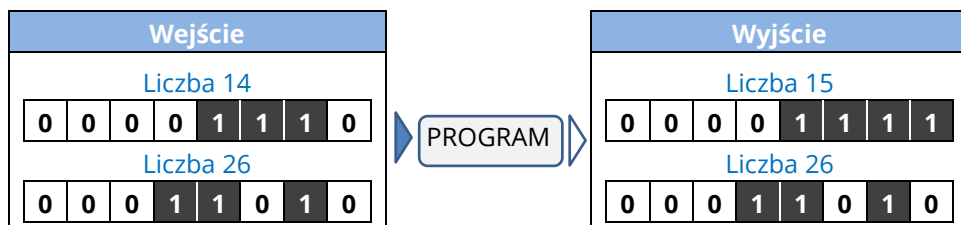
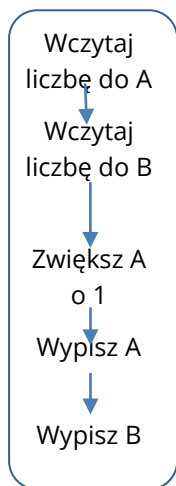


Diagram rozwiązania i jego implementacja.



Mamy wczytać dwie liczby. Zadanie precyzuje pudełka, do których wczytujemy liczby, dlatego wczytujemy najpierw do A, a następnie do B. Korzystamy z instrukcji „Wczytaj”, wybierając pudełka A i B.

Wczytaj do	A
Wczytaj do	B

Mamy wypisać pierwszą liczbę zwiększoną o 1. Najpierw zwiększamy A o 1, a następnie wypisujemy nową wartość A.

Zwiększ	A	o	1
Wypisz pudełko	A		

Na końcu wypisujemy liczbę z pudełka B.

Wypisz pudełko	B
----------------	---

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.

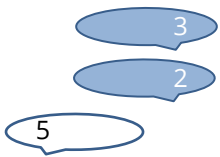
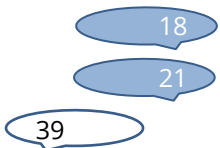
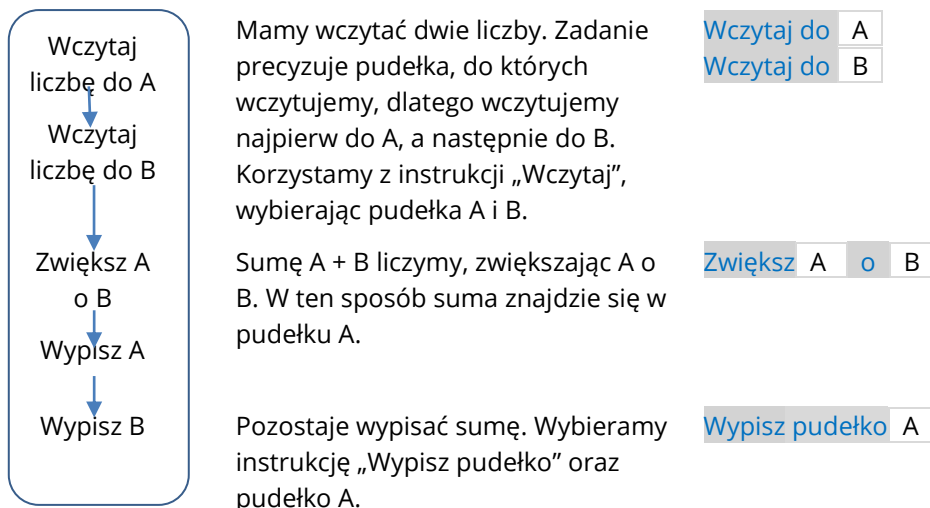
Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 3 i 2 program powinien wypisać na wyjściu 5 (bo $3 + 2 = 5$).	
Dla podanych na wejściu liczb 18 i 21 program powinien wypisać na wyjściu 39 (bo $18 + 21 = 39$).	

Diagram rozwiązania i jego implementacja.



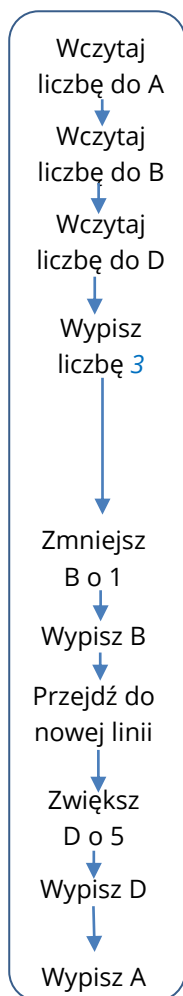
Przykład 3.

Napisz program, który wczyta liczby do pudełek A, B, D i wypisze liczbę 3 oraz B zmniejszone o 1. Następnie w nowej linii program 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 3 oraz 1 (bo $2 - 1 = 1$) i w nowej linii 6 (bo $1 + 5 = 6$) oraz 3.	3 2 1 31 63

Diagram rozwiązania i jego implementacja.



Wczytujemy liczby do pudełek A, B i D. Czy wiesz, dlaczego istotne jest zachowanie 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”.

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.

Na końcu wypisujemy pudełko A.

Wczytaj do A

Wczytaj do B

Wczytaj do D

Wypisz napis 3

Zmniejsz B o 1

Wypisz pudełko B

Przejdź do nowej linii

Zwiększ D o 5

Wypisz pudełko D

Wypisz pudełko A



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

Zadania InstaKod 5. Operacje arytmetyczne 3.

Poziom 1

- IV 5.1 Napisz program, który wczyta liczby do pudełek A i B, następnie wypisze B oraz A zmniejszone 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 wypisze wczytaną wartość.
- IV 5.5 Napisz program, który wczyta liczbę do pudełka B, wypisze tę liczbę powiększoną o 3 oraz liczbę 5. Następnie wczyta liczbę do pudełka C i wypisze C.
- IV 5.6 Napisz program, który wczyta liczby do pudełek A, B i D, 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, doda do każdej 1, a następnie wypisze je w oddzielnych wierszach w kolejności wczytania.
- IV 5.8 Napisz program, który wczyta liczby do pudełek A, B i C i wypisze ich sumę.
- IV 5.9 Napisz program, który wczyta liczby do pudełek A, B i C i wypisze sumę A i B zmniejszoną o C.
- IV 5.10* Napisz program, który wczyta liczby do pudełek C, A i B, wypisze sumę A i B oraz dwukrotnie wypisze C.

Instakod 6

Operacje arytmetyczne 4

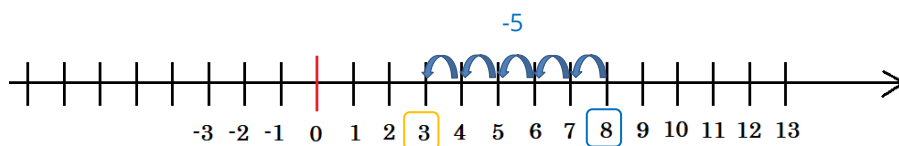
Zwiększanie i zmniejszanie wartości pudełka o wartość podanego pudełka. Krok dalej.

Wiesz już, jak policzyć sumę dwóch wczytanych liczb. Jeśli liczby znajdują się w pudełkach A i B, możemy zrobić to na dwa sposoby:

- **Zwiększ** A **o** B policzy sumę $A + B$ i zapisze wynik w pudełku A.
- **Zwiększ** B **o** A policzy sumę $B + A$ i zapisze wynik w pudełku B.

Dowiemy się teraz, w jaki sposób policzyć **różnicę** dwóch wczytanych liczb.

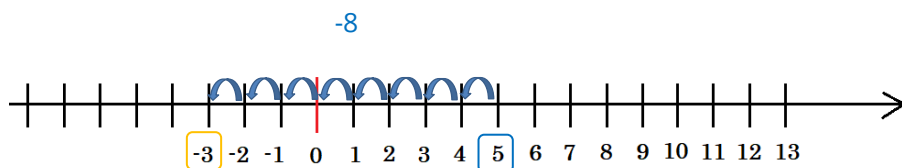
Zilustrujmy operację $A - B$ na osi liczbowej dla liczb $A = 8$ i $B = 5$. Zaczynamy w 8 i przesuwamy się o 5 kresek w lewo do liczby 3.



Poniższy program zapisuje różnicę $A - B$ w pudełku A. Po uruchomieniu powyższego programu i wykonaniu instrukcji **Zmniejsz** A **o** B liczba w pudełku A zostanie zaktualizowana do 3 (bo $8 - 5 = 3$). Liczba w pudełku B nie zostanie zmieniona.

Pudełka	Edytor	Konsola
A: 3 B: 5 C: 0 D: 0	Wczytaj do A Wczytaj do B Zmniejsz A o B Wypisz pudełko A	8 5 3

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 policzona różnica $B - A$. Zauważmy, że teraz w B pojawiła się różnica, to jest $5 - 8$, czyli -3, a w A pozostała wczytana wartość 8.



Suma i różnica dwóch liczb

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

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

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.

Zmniejsz B o 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 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.

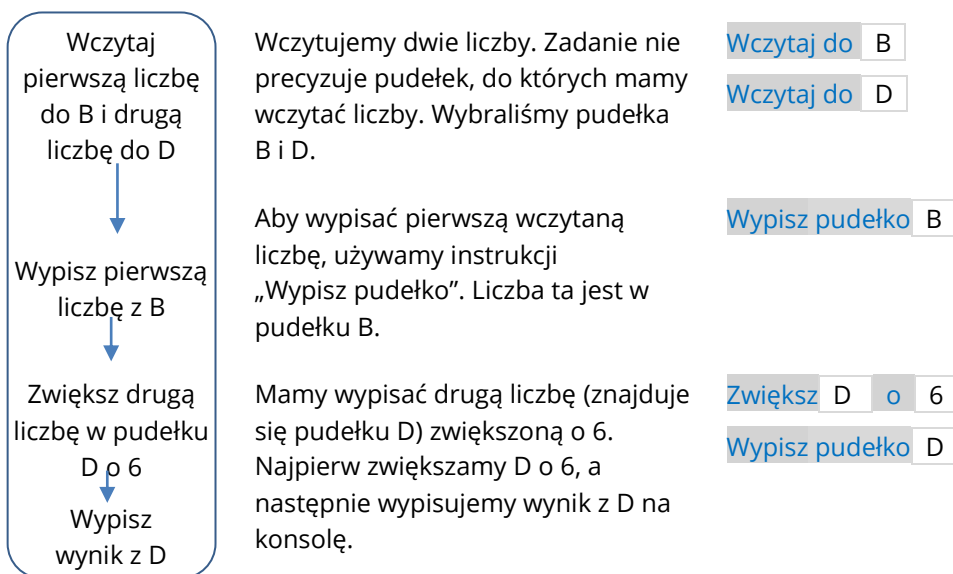
Wejście i wyjście

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

Konsola

Diagram illustrating the console interaction. It shows two input bubbles containing the numbers 2 and 5, and one output bubble containing the number 211.

Diagram rozwiązania i jego implementacja.



Przykład 2.

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

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

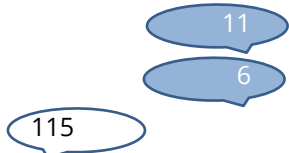
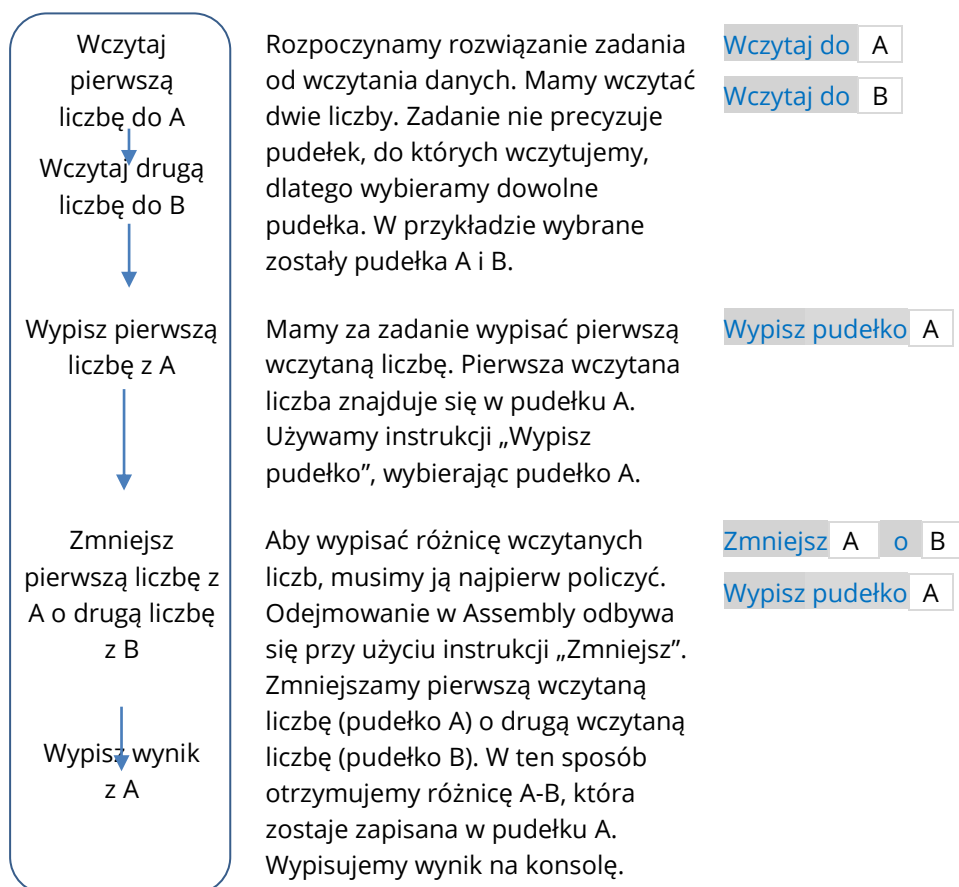
Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 11 i 6, program powinien wypisać na wyjściu 11 i 5 (bo $11 - 6 = 5$).	

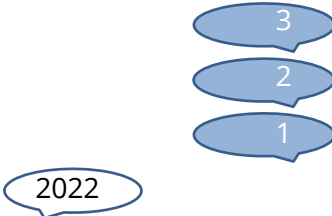
Diagram rozwiązania i jego implementacja.



Przykład 3.

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

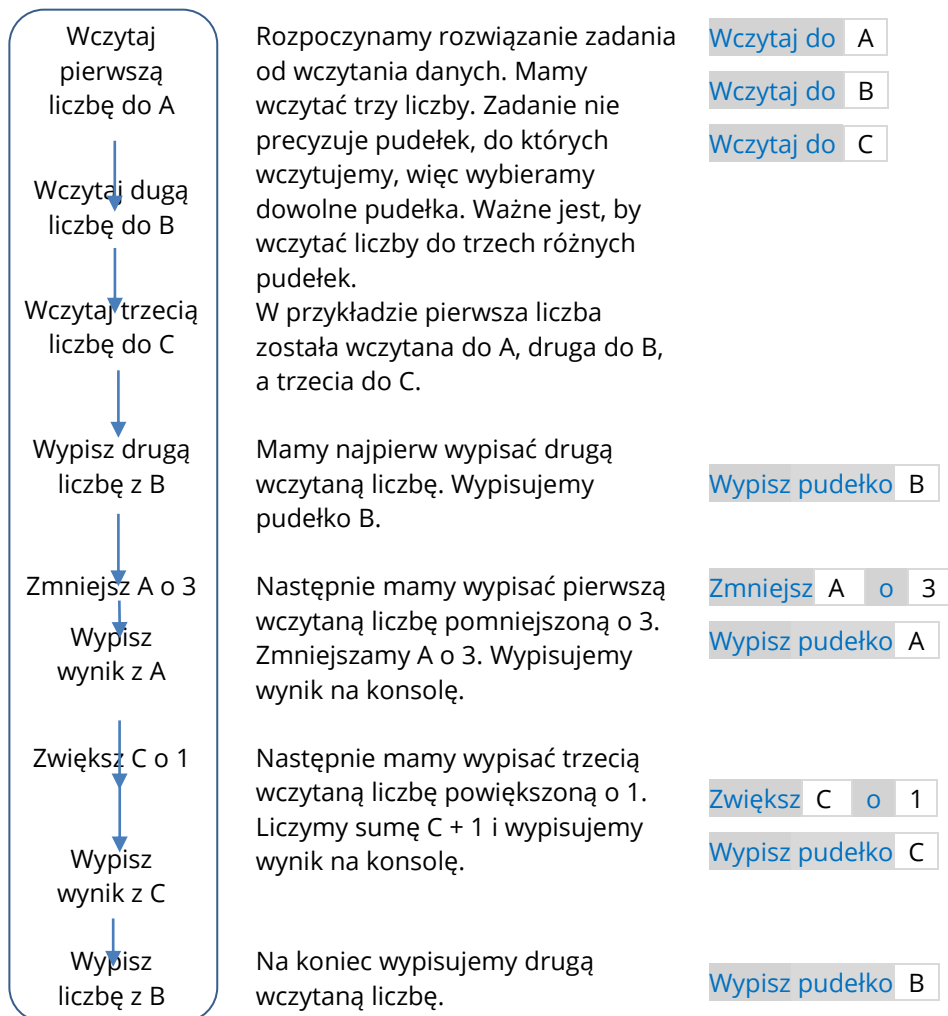
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

Wejście i wyjście	Konsola
Dla podanych przez użytkownika liczb: 3, 2 i 1 program powinien wypisać: 2, następnie 0 (bo $3 - 3 = 0$), 2 (bo $1 + 1 = 2$), 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><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><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><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.



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

Zadania. InstaKod Rozdział 6. Operacje arytmetyczne 4.

Poziom 1

- IV 6.1 Napisz program, który wczyta dwie liczby i wypisze 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 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 drugą liczbę.
- IV 6.5 Napisz program, który wczyta dwie liczby i wypisze drugą powiększoną o 2 oraz wypisze dwukrotnie pierwszą pomniejszoną o 2.
- IV 6.6 Napisz program, który wczyta dwie liczby i wypisze napis 7b. Następnie przejdzie do nowej linii i 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 pierwszą, drugą zmniejszoną o 3 i trzecią zwiększoną o 1.
- IV 6.8 Napisz program, który wczyta trzy liczby i wypisze różnicę pierwszej i drugiej powiększoną o trzecią.
- IV 6.9 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 6.10* Napisz program, który wczyta trzy liczby, zwiększy każdą z nich o 3, a następnie wypisze je w osobnych wierszach.
- IV 6.11* Napisz program, który wczyta trzy liczby i wypisze w osobnych wierszach: trzecią zmniejszoną o 3, liczbę 8, drugą zwiększoną o 2 i pierwszą zwiększoną o 5.

InstaKod 7

Suma i różnica 1

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

Do policzenia sumy liczb zapisanych w pudełkach A i B korzystamy z instrukcji **Zwiększ** A ☐ B.

Zadanie

Policz sumę dwóch wczytanych liczb.

Wykorzystamy poniższy program. Po uruchomieniu wczytajmy liczby 8 i 2. Pierwsza trafi do pudełka A, a druga do pudełka B. Po wykonaniu operacji **Zwiększ** A ☐ B do pudełka A trafi suma $8 + 2 = 10$.

The screenshot shows the InstaKod 7 interface with three panels: Pudełka, Edytor, and Konsola. In the Pudełka panel, A is 10, B is 2, C is 0, and D is 0. In the Edytor panel, the program consists of four steps: 1. Wczytaj do A, 2. Wczytaj do B, 3. Zwiększ A o B, and 4. Wypisz pudełko A. In the Konsola panel, the input values 8 and 2 are shown, and the output is 10.

Zwróć uwagę, że w ten sposób tracimy informację o liczbie, która została wczytana do pudełka A. W niektórych zadaniach ta informacja będzie nam potrzebna. Wtedy będziemy musieli zastosować inne podejście.

Sposób 1.

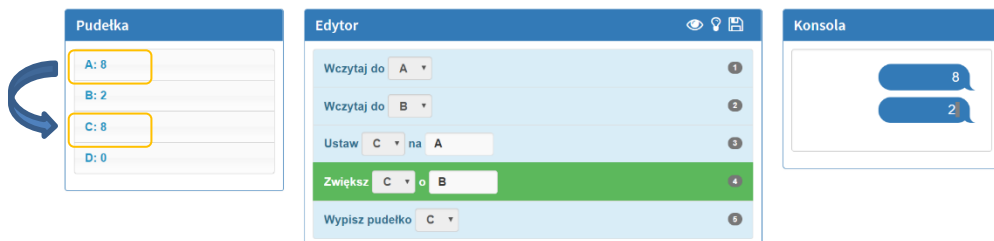
Możemy odtworzyć liczbę wczytaną do A, wykonując odwrotną operację. W tym przypadku będzie to **Zmniejsz** A ☐ B. Program wykona działanie $10 - 2 = 8$ i zapisze wynik w pudełku A.

The screenshot shows the InstaKod 7 interface with three panels: Pudełka, Edytor, and Konsola. In the Pudełka panel, A is 8, B is 2, C is 0, and D is 0. In the Edytor panel, the program consists of six steps: 1. Wczytaj do A, 2. Wczytaj do B, 3. Zwiększ A o B, 4. Wypisz pudełko A, 5. Zmniejsz A o B, and 6. Wypisz pudełko A. In the Konsola panel, the input values 8 and 2 are shown, and the output is 10.

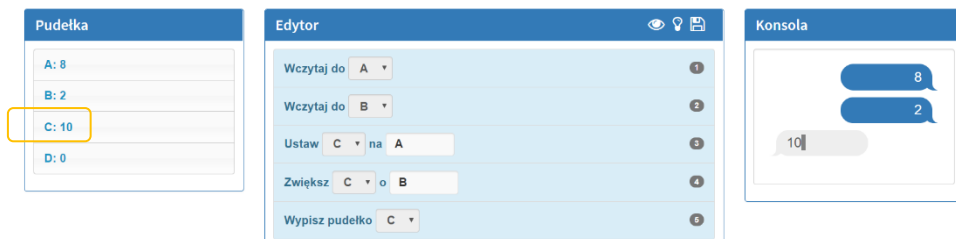
Sposób 2

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

Przepiszmy do pudełka C liczbę z pudełka A, korzystając z instrukcji **Ustaw C na A**. Do pudełka C trafi liczba 8.



Następnie policzymy sumę $A + B$ w pudełku C. Korzystamy z instrukcji **Zwiększ C o B**. W ten sposób policzyliśmy sumę $A + B$, pozostawiając niezmienione liczby w pudełkach A i B.



Pudełko pomocnicze

Pudełko pomocnicze służy do liczenia sumy lub różnicy dwóch innych pudełek. 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 poizsze 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 jest 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

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 znajdującą się w pudełku A oraz liczby znajdującą się w pudełku B i zapisuje wynik w pudełku A.

Przypisanie wartości do pudełka

Instrukcja **Ustaw** przypisuje (wstawia) do wybranego pudełka konkretną liczbę lub wartość 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 i B będą znajdować się te same liczby.

Przykład 1.

Napisz program, który wczyta dwie liczby do pudełek A i D, wyliczy ich sumę w pudełku pomocniczym C i wypisze wynik. Następnie 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 (bo $6 + 8 = 14$) oraz 6 i 8.

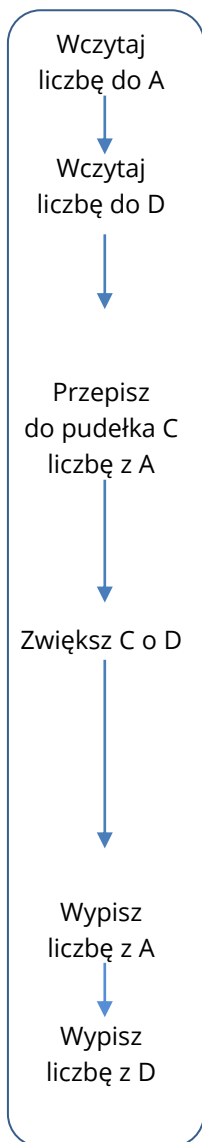
Konsola

The diagram shows a console interaction. On the right, two blue speech bubbles contain the numbers 6 and 8, representing input. On the left, a blue speech bubble contains the number 1468, representing the output.

Chcemy policzyć sumę dwóch liczb wczytanych do pudełek A i D, nie zmieniając ich wartości. Sumę obliczamy w pudełku pomocniczym C.

Pudełko A: 6	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	1	1	0	Pierwsza wczytana liczba
0	0	0	0	0	1	1	0			
Pudełko B: 0	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0			
Pudełko C: 14	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	0	0	0	0	1	1	1	0	Pudełko pomocnicze Suma A + D, czyli 14
0	0	0	0	1	1	1	0			
Pudełko D: 8	<table><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			

Diagram rozwiązania i jego implementacja.



Rozpoczynamy rozwiązanie zadania od wczytania danych. Mamy wczytać dwie liczby. Zadanie precyzuje pudełka, do których wczytujemy, dlatego wybieramy pudełka A i D. Używamy instrukcji „Wczytaj” oraz odpowiednich pudełek.

Wczytaj do A

Wczytaj do D

Przechodzimy do wykonywania operacji arytmetycznych. Mamy do policzenia sumę wczytanych liczb. Sumę liczymy zwiększając wartość jednej liczby o drugą.

Ustaw C na A

Zwiększ C o D

Zwróć uwagę, że po wypisaniu sumy mamy jeszcze wypisać wczytane liczby. Jeśli zwiększymy A o D, to utracimy informację o liczbie wczytanej do A. Sumę policzymy więc w pudełku pomocniczym C. Najpierw do C przepisujemy pierwszą wczytaną liczbę, która jest w A. Następnie zwiększymy ją o D.

Możemy teraz przejść do wypisania wyniku oraz wczytanych liczb. Wypisujemy pudełko C, gdzie przechowujemy policzoną sumę. Następnie wypisujemy pudełko A, w którym jest pierwsza wczytana liczba oraz pudełko D, w którym jest druga wczytana liczba.

Wypisz pudełko C

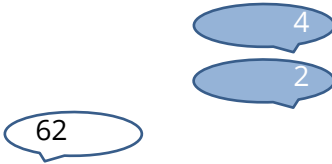
Wypisz pudełko A

Wypisz pudełko D

Przykład 2.

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

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

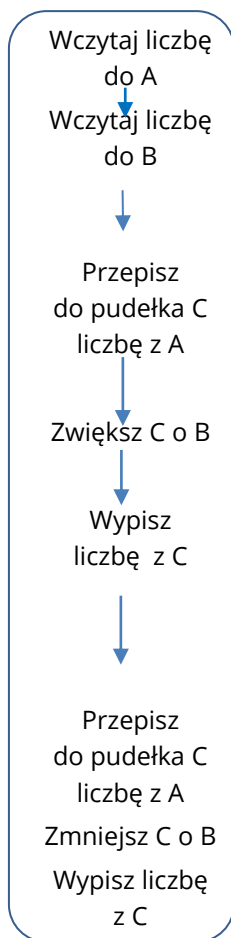
Wejście i wyjście	Konsola
Dla podanych na wejściu liczb 4 i 2 program powinien wypisać na wyjściu 6 (bo $4 + 2 = 6$) oraz 2 (bo $4 - 2 = 2$).	

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

Pudełko A: 4	<table><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	Pierwsza wczytana liczba
0	0	0	0	0	1	0	0			
Pudełko B: 2	<table><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: 6	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	1	1	0	Pudełko pomocnicze Suma A + B, czyli 6
0	0	0	0	0	1	1	0			
Pudełko C: 2	<table><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			

Pudełko C wykorzystujemy dwukrotnie. Po obliczeniu sumy wypiszemy ją i nie będzie już nam potrzebna. Skorzystamy z pudełka C drugi raz, by policzyć różnicę.

Diagram rozwiązania i jego implementacja.



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

Wczytaj do A

Wczytaj do B

W tym zadaniu mamy do policzenia zarówno sumę, jak i różnicę. Aby policzyć 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.

Ustaw C na A

Zwiększ C o B

Wypisz pudełko C

Drugi raz korzystamy z pudełka C, tym razem do policzenia 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



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

Zadania. InstaKod Rozdział 7. Suma i różnica 1.

Poziom 1

- IV 7.1 Napisz program, który wczyta dwie liczby do pudełek A i B, wyliczy ich różnicę w pudełku pomocniczym i wypisze wynik. Następnie wypisze wczytane liczby.
- IV 7.2 Napisz program, który wczyta dwie liczby do pudełek A i D, wyliczy ich sumę w pudełku pomocniczym i wypisze wynik. Następnie wypisze wczytane liczby.
- IV 7.3 Napisz program, który wczyta dwie liczby do pudełek A i B, wyliczy ich różnicę w pudełku pomocniczym, wypisze liczbę 3 oraz wyliczoną różnicę.

Poziom 2

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

Poziom 3

- IV 7.7 Napisz program, który wczyta dwie liczby do pudełek A i B, wyliczy ich sumę w pudełku pomocniczym i wypisze wynik. Następnie w nowej linii dwukrotnie wypisze pierwszą wczytaną liczbę.
- IV 7.8 Napisz program, który wczyta dwie liczby do pudełek A i B, wyliczy w pudełkach pomocniczych ich sumę i różnicę. Następnie wypisze różnicę i sumę w osobnych wierszach.
- IV 7.9 Napisz program, który wczyta dwie liczby do pudełek A i B oraz wyliczy ich różnicę w pudełku pomocniczym. Następnie wypisze liczbę 3, różnicę zwiększoną o 5 oraz wczytane liczby.

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.

Założmy, że chcemy wczytać dwie liczby i policzyć ich sumę.

Zaplanujmy, gdzie zapiszemy liczby i gdzie wykonamy działania.

	Sposób I	Sposób II
	Możemy wczytać liczby do A i B, a obliczenia wykonać w pudełku C.	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, wypisze ich różnicę oraz wczytane liczby.

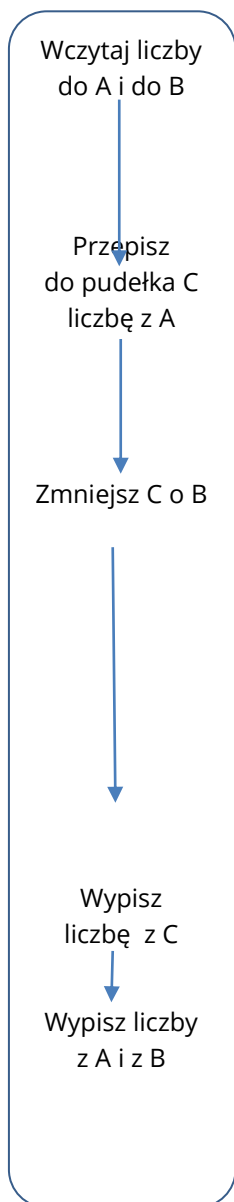
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

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

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

Pudełko A: 10	<table><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><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><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 policzenia 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 policzona różnica. Następnie wypisujemy pudełko A, w którym jest pierwsza wczytana liczba oraz pudełko B, w którym jest druga wczytana liczba.

Wypisz pudełko C

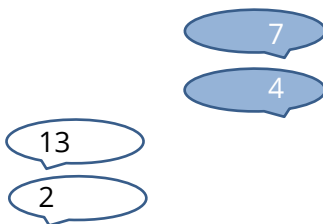
Wypisz pudełko A

Wypisz pudełko B

Przykład 2.

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

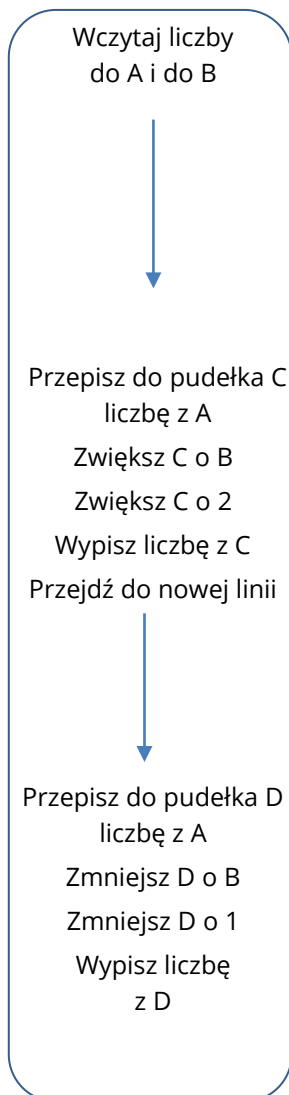
Oto przykładowe wejście i wyjście oraz zapis interakcji użytkownika z komputerem dla tych danych na konsoli.

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

Mamy policzyć 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><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><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><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><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 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 policzenia $A + B + 2$. Przepisujemy do C wartość z A, korzystając z instrukcji

Ustaw C na A

Zwiększ C o B

Zwiększ C o 2

Wypisz pudełko C

Przejdź do nowej linii

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

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

Użyjemy pudełka pomocniczego D do policzenia różnicy wczytanych liczb pomniejszonej o 1. Przepisujemy do D wartość z A, korzystając z instrukcji

Ustaw D na A

Zmniejsz D o B

Zmniejsz D o 1

Wypisz pudełko D

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



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

Zadania InstaKod 8. Suma i różnica 2.

Poziom 1

- IV 8.1 Napisz program, który wczyta dwie liczby i wypisze ich sumę oraz liczby w kolejności wczytania.
- IV 8.2 Napisz program, który wczyta dwie liczby i wypisze ich różnicę powiększoną o 1 oraz drugą wczytaną liczbę.
- IV 8.3 Napisz program, który wczyta dwie liczby, a następnie 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 wypisze dwukrotnie pierwszą wczytaną liczbę.
- IV 8.5 Napisz program, który wczyta dwie liczby, wypisze 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 wypisze w osobnych wierszach ich sumę i różnicę.
- IV 8.8 Napisz program, który wczyta dwie liczby, a następnie wypisze napis *abc*, sumę wczytanych liczb pomniejszoną o 1 oraz ich różnicę.
- IV 8.9 Napisz program, który wczyta dwie liczby i wypisze drugą zwiększoną o 7, sumę wczytanych liczb oraz pierwszą wczytaną liczbę.

InstaKod 9

Instrukcja warunkowa 1

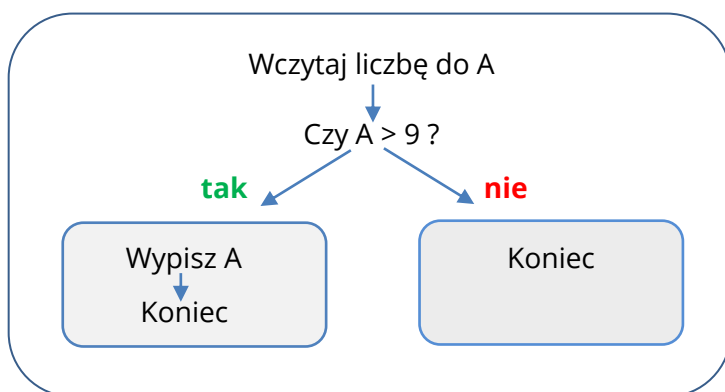
Pierwsze starcie. Jeśli warunek jest spełniony, to podejmij działanie, w przeciwnym razie zakończ program.

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

Przykład

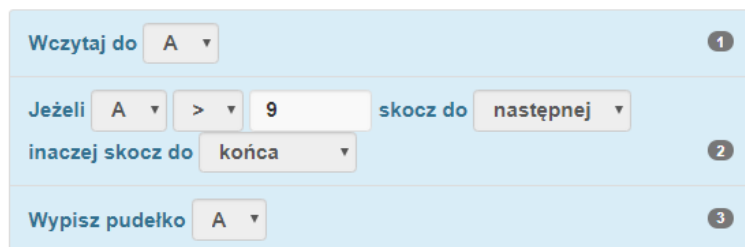
Napisz program, który wypisze wczytaną liczbę, jeśli jest większa od 9.

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

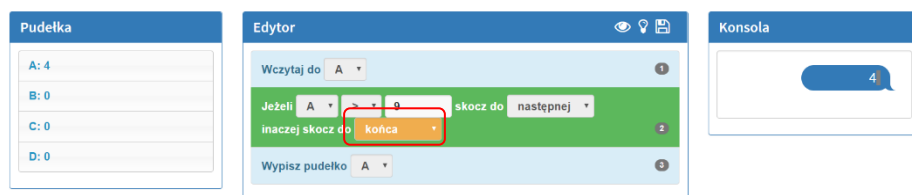


Do sprawdzania wszelkich porównań używamy w Assembly instrukcji „Jeżeli”. Dla powyższego przykładu porównaniem jest $A > 9$. Jeśli **warunek jest spełniony**, przechodzimy (skaczemy) do następnej linii kodu, gdzie wypisujemy wartość pudełka A i kończymy program. Jeśli **warunek nie jest spełniony**,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

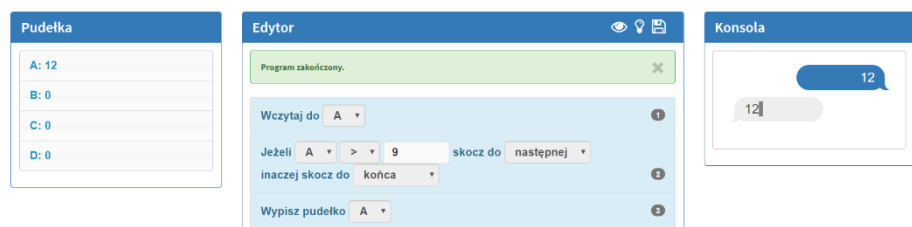
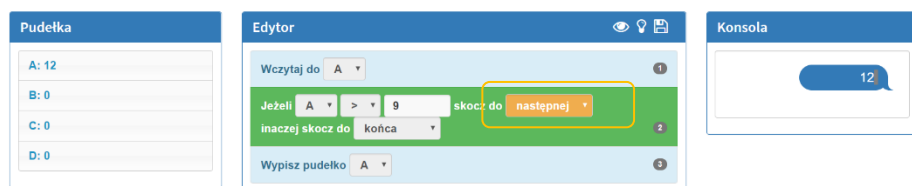
Rozwiązanie przeniesione z diagramu do edytora Assembly będzie wyglądało następująco:



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.



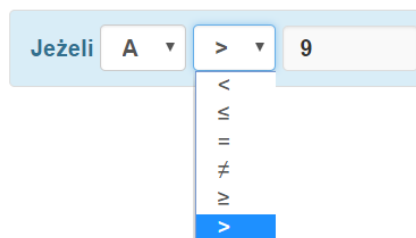
Instrukcja **Jeżeli** składa się z trzech osobnych części:

1	2	3
Jeżeli A > 9	skocz do następnej	inaczej skocz do końca
Warunek	Informacja, do której linii kodu program ma skoczyć, jeśli warunek jest spełniony.	Informacja, do której linii kodu program ma skoczyć, jeśli warunek nie jest spełniony.

W Assembly warunkiem jest porównanie jednego z czterech pudełek z innym pudełkiem, bądź liczbą.

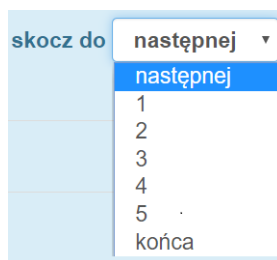
Dostępne są następujące porównania:

- < 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.



W przykładzie powyżej pytamy: „Czy A jest większe od 9?”

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

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

Kiedy warunek $A > 9$ nie jest spełniony?

Nie jest spełniony jest 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 warunkowe wykonanie części programu.

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

jest instrukcją warunkową.

Jeżeli warunek $A > 9$ **jest** spełniony, to wykonana zostanie instrukcja z następnej linii kodu.

Jeżeli warunek **nie jest** spełniony (gdy $A < 9$ 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 1.

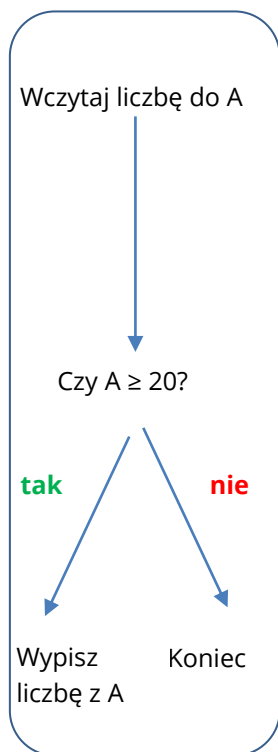
Napisz program, który wczyta liczbę i sprawdzi, czy jest większa lub równa 20. Jeśli tak, to wypisze ją na ekran.

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.	

Na pytanie „Czy $A \geq 20$?” 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 **drzewkiem**, a jego strzałki **gałęziami**.

Diagram rozwiązania i jego implementacja.



Wczytujemy dane. Zadanie nie precyzuje pudełka, do którego wczytujemy liczbę. W przykładzie wybrane zostało pudełko A.

Sprawdzamy, czy wczytana liczba jest większa lub równa 20. Do wszelkich porównań w Assembly używamy instrukcji „Jeżeli”.

Zapisujemy warunek $A \geq 20$. Jeśli warunek jest spełniony, to skaczemy do następnej linii kodu i podejmujemy dalsze działanie. W przeciwnym przypadku skaczemy do końca.

Wczytaj do A

Jeżeli $A \geq 20$ skocz
do następnej
inaczej skocz
do końca

Pozostało jedynie wypisać wartość pudełka A.

Wypisz pudełko A

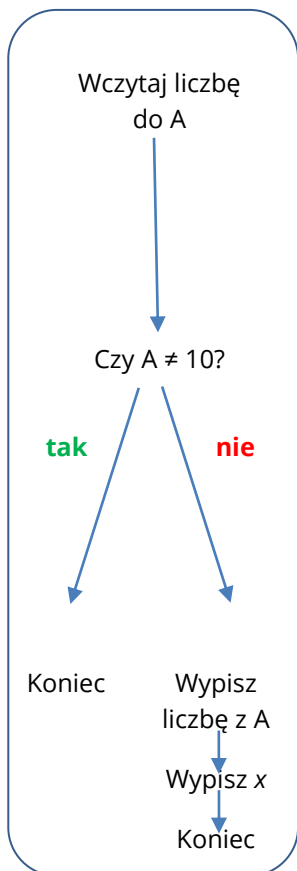
Przykład 2.

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 (bo $10 - 1 = 9$) oraz literę x.	10 9x
Dla podanej na wejściu liczby 14 program powinien sprawdzić, że warunek $14 \neq 10$ jest spełniony i zakończyć działanie.	14

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 $A \neq 10$
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 $A = B$. Jeśli $A = B$, skaczemy do następnej linii kodu i wykonujemy dalszą część programu. Jeśli $A \neq B$, kończymy program.

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

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

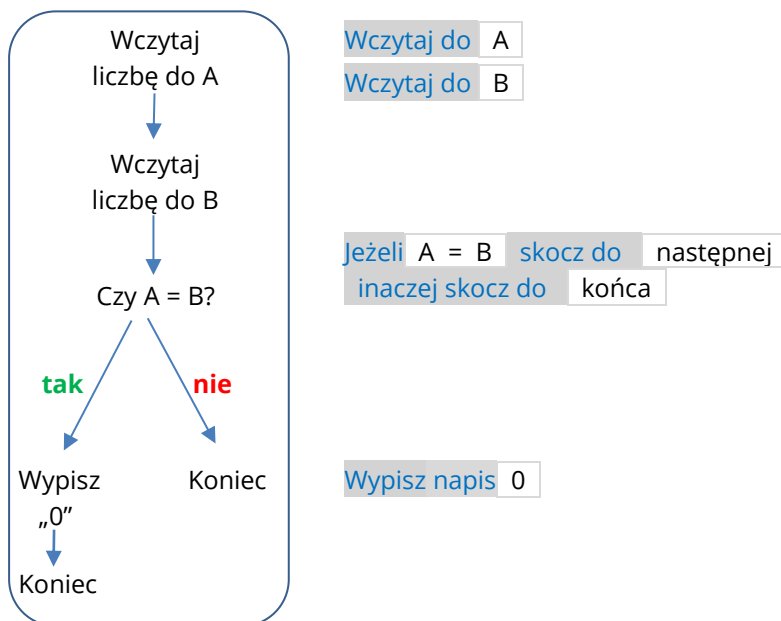
Zmniejsz A o B

Wypisz pudełko A

Rozwiązaliśmy zadanie poprawnie. Ale być może ktoś z Was już zauważył, że to zadanie ma inne, dużo prostsze rozwiązanie.

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

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



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



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

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 drugiej. Jeśli tak, to wypisze ich sumę.
- IV 9.8 Napisz program, który wczyta dwie liczby i sprawdzi, czy pierwsza jest większa od 7. Jeśli tak, to wypisze pierwszą i drugą zwiększoną o 2.
- IV 9.9 Napisz program, który wczyta dwie liczby i jeśli druga zwiększona o 3 jest mniejsza lub równa pierwszej, to wypisze *Tak*.
- IV 9.10* Napisz program, który wczyta dwie liczby i jeśli ich suma jest większa od 12, to wypisze wczytane liczby w oddzielnych wierszach.

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.

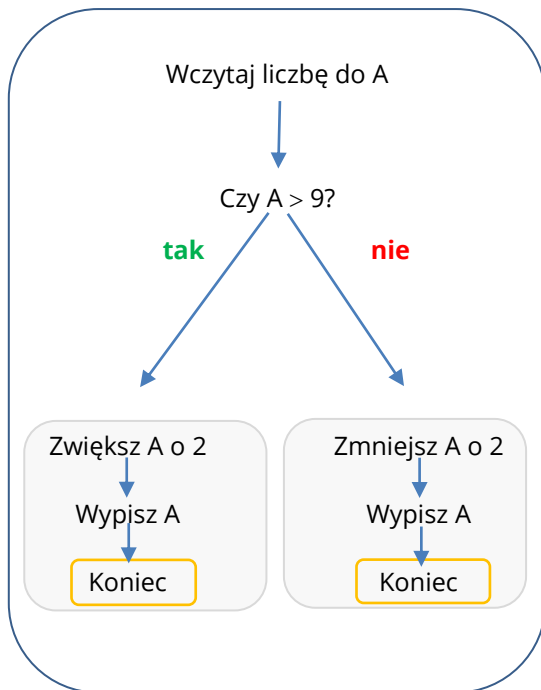
Przykład 1.

Napisz program, który wczyta liczbę oraz sprawdzi, czy jest ona większa od 9. Jeśli tak, to doda do niej 2 i wypisze wynik, a 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 (bo $12 + 2 = 14$).	
Dla podanej na wejściu liczby 5 program powinien sprawdzić, że warunek $5 > 9$ nie jest spełniony i wypisać na wyjściu 3 (bo $5 - 2 = 3$).	

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.

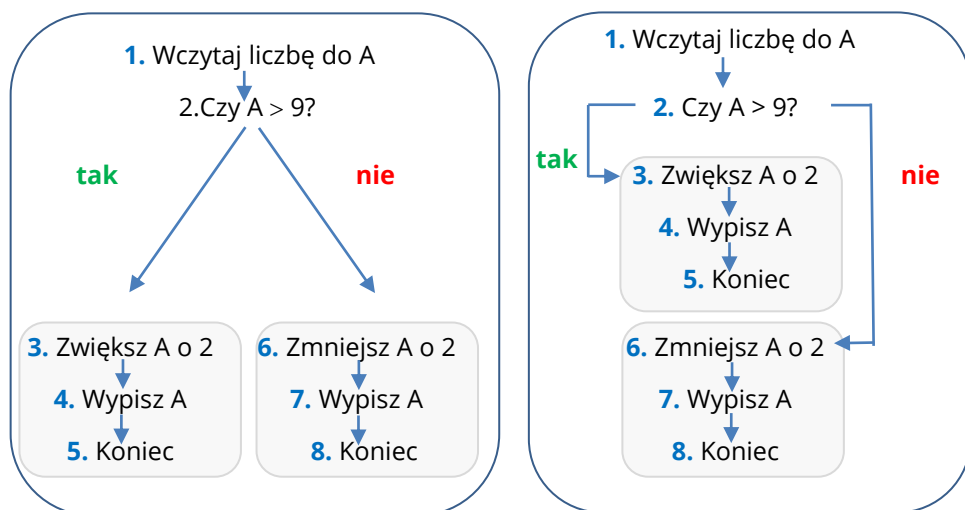
Jeśli odpowiedzią jest **nie**, liczymy różnicę wczytanej liczby i liczby 2. Następnie wypisujemy wynik na konsolę.

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 drzewka musimy przetłumaczyć na **diagram liniowy**.

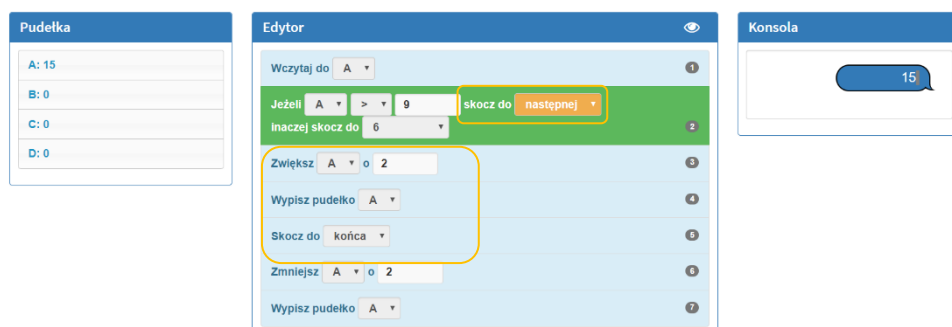
W tym celu ponumerujemy kroki w diagramie drzewkowym.

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

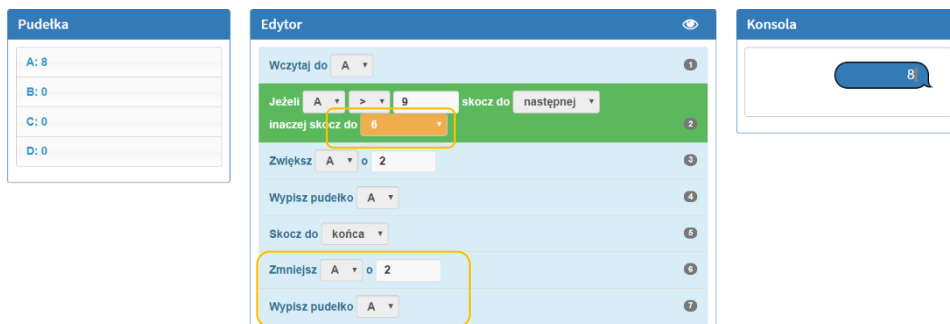


Zwróć uwagę, że ostatni krok „8. Koniec” nie jest potrzebny na końcu programu.

Zaimplementujemy diagram liniowy w edytorze Assembly. Następnie uruchommy nasz program. Zrobmy 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.



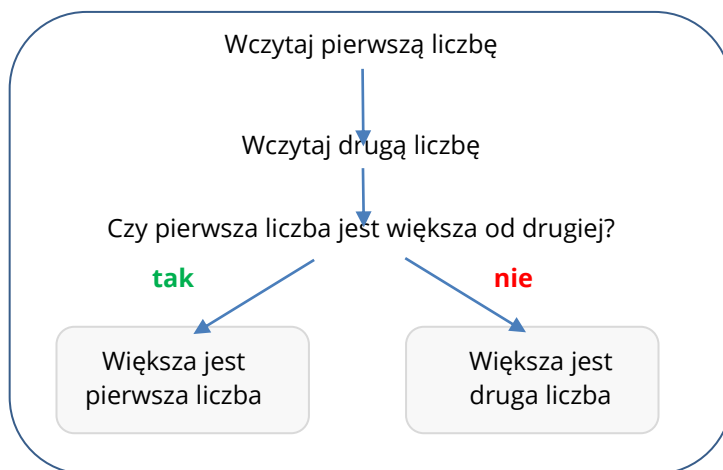
Dla wczytanej liczby 8, warunek $8 < 9$ **nie jest** spełniony. Wykonany zostanie blok z linii 6-7. Liczba w pudełku A zostanie zmniejszona o 2 i na konsolę zostanie wypisany wynik. Następnie program zakończy działanie.



Przykład 2.

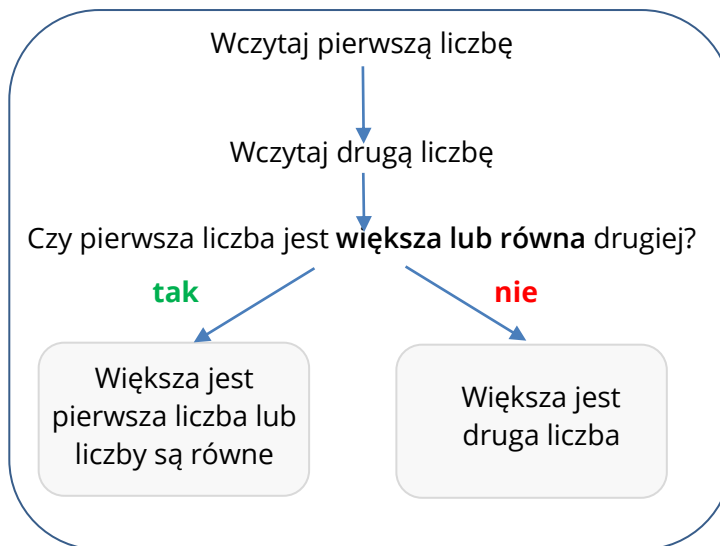
Napisz program, który wczyta dwie liczby i wypisze je najpierw większą, potem mniejszą. Jeśli wczytane liczby są równe, program wypisze je obok siebie.

Zapiszmy diagram rozwiązania dla tego zadania. Mamy dwie liczby i sprawdzamy, czy pierwsza jest większa od drugiej. Jeśli tak, to większa jest pierwsza, a jeśli nie, to większa jest druga liczba.



- Sprawdźmy działanie powyższego algorytmu dla liczb 12 i 8.
Czy 12 jest większe od 8? Tak, więc pierwsza liczba jest większa.
Algorytm działa poprawnie.
- Sprawdźmy działanie powyższego algorytmu dla liczb 5 i 7.
Czy 5 jest większe od 7? Nie, więc druga liczba jest większa.
Algorytm działa poprawnie.
- Sprawdźmy działanie powyższego algorytmu dla liczb 10 i 10.
Czy 10 jest większe od 10? Nie. Według powyższego diagramu większą liczbą będzie 10. A nie jest to prawdą, bo liczby są równe.

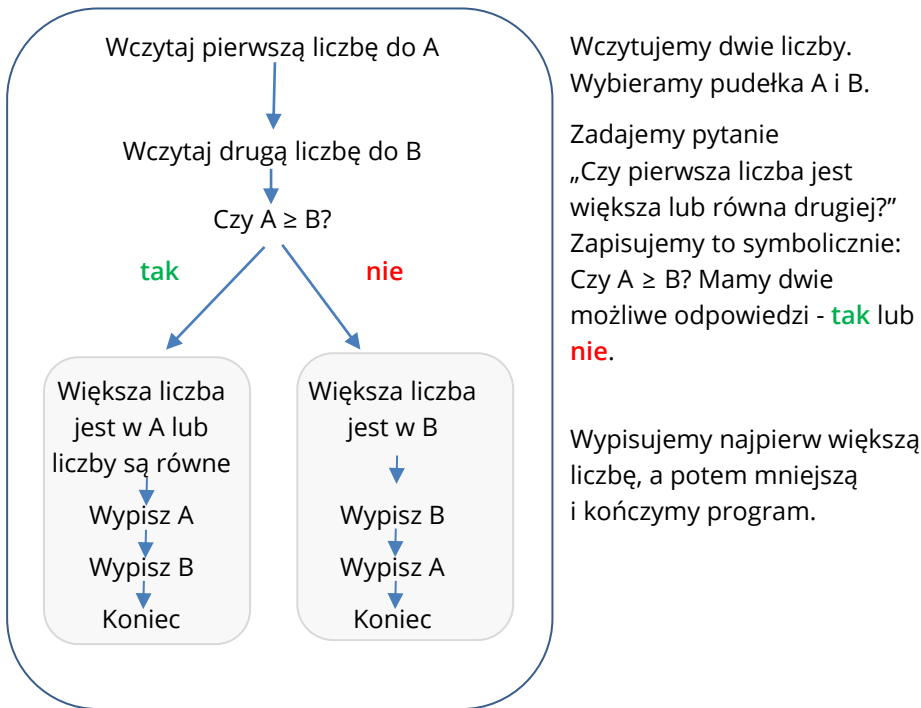
Musimy więc poprawić diagram algorytmu. Zmieńmy pytanie na:
„Czy pierwsza liczba **jest większa lub równa** drugiej?”
Jeśli warunek jest spełniony to większa jest pierwsza liczba lub liczby są równe.



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.	12 8 128
Dla podanych na wejściu liczb 3 i 7 program powinien wypisać na wyjściu najpierw 7, a następnie 3.	3 7 73
Dla podanych na wejściu liczb 6 i 6 program powinien wypisać na wyjściu dwa razy liczbę 6 obok siebie.	6 6 66

Diagram rozwiązania i jego implementacja.

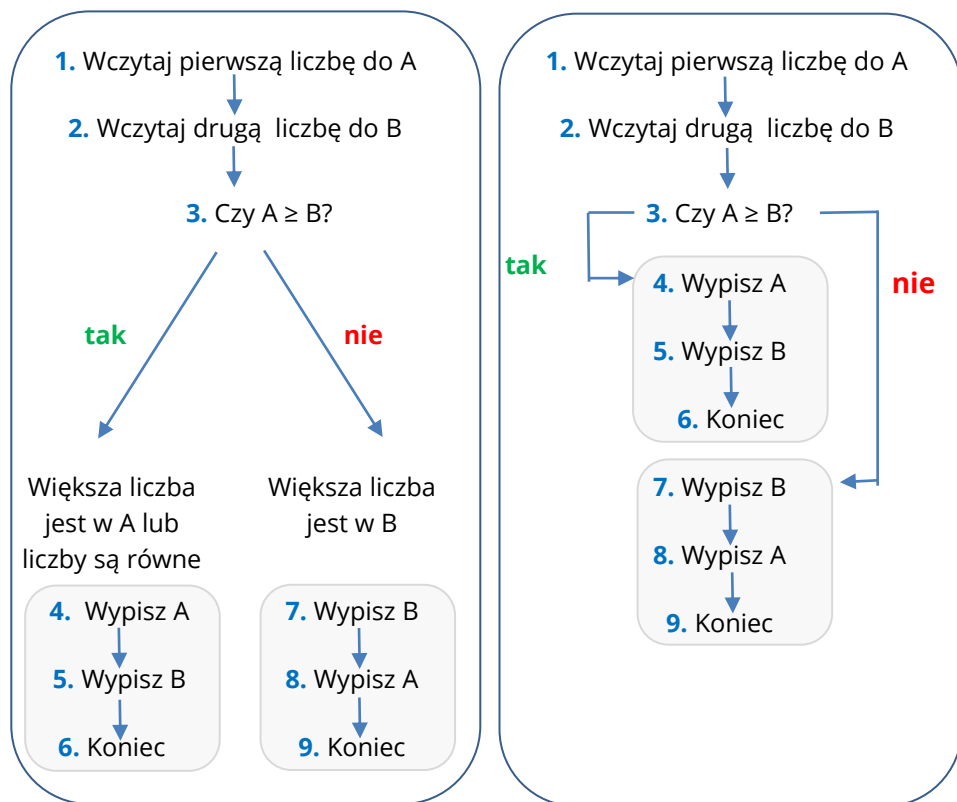


Aby zaimplementować rozwiązanie, musimy je przepisać na kod linii po linii.

W tym celu tłumaczymy **diagram w formie drzewka** na **diagram liniowy**.

Numerujemy kroki w diagramie drzewkowym.

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



Zwróć uwagę, że ostatni krok „9. Koniec” nie jest potrzebny na końcu programu.

Teraz z łatwością przepisujemy diagram liniowy na kod:

Wczytaj do	A ▾	1				
Wczytaj do	B ▾	2				
Jeżeli	A ▾	≥ ▾	B	skocz do	następnej ▾	3
inaczej skocz do	7 ▾					
Wypisz pudełko	A ▾	4				
Wypisz pudełko	B ▾	5				
Skocz do	końca ▾	6				
Wypisz pudełko	B ▾	7				
Wypisz pudełko	A ▾	8				



instaKod

Przejdź do zeszytu ćwiczeń i wypełnij karty pracy, nie korzystając z komputera.

Zaloguj się na swoje konto na platformie InstaKod i rozwiąż zadania. Diagramy wymyślonych przez siebie rozwiązań zapisuj w zeszycie do informatyki.

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 pierwszą i drugą. W przeciwnym razie wypisze 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 są równe. Jeśli tak, wypisze ich różnicę. W przeciwnym razie wypisze *Nie*.
- IV 10.8 Napisz program, który wczyta dwie liczby i wypisze większą dwukrotnie w oddzielnych wierszach.
- IV 10.9 Napisz program, który wczyta dwie liczby i wypisze mniejszą, a potem większą.
- IV 10.10* Napisz program, który wczyta dwie liczby i wypisze większą zmniejszoną o 1 oraz mniejszą.

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 do programu

Wejście to dane podane do programu.

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.

Wyjście z 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 jest **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

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

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

Przypisanie wartości do pudełka

Instrukcja **Ustaw** przypisuje (wstawia) do wybranego pudełka konkretną liczbę lub wartość 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 i B będzie znajdować się ta sama liczba.

Instrukcja warunkowa

Instrukcja warunkowa umożliwia warunkowe wykonanie części programu.

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.

TECHNOLOGIE INFORMACYJNE

Ucząc się, komunikując się ze znajomymi, czy też rozwijając swoje hobby, korzystasz z wielu różnych aplikacji komputerowych. W tym dziale przyjrzymy się bliżej 4 aplikacjom, które pomogą ci przy realizowaniu różnorodnych zadań z polskiego, historii, przyrody i innych przedmiotów szkolnych. Są to:

1. program graficzny,
2. edytor tekstu,
3. program do prezentacji,
4. arkusz kalkulacyjny.

Omówione zostaną wybrane funkcje tych programów. W każdym rozdziale znajdziesz przygotowane specjalnie dla Ciebie ćwiczenia wspomagające korzystanie ze wspomnianych aplikacji. Oprócz tego przygotowane zostały proponowane tematy projektów na inne przedmioty szkolne, do realizacji których możesz wykorzystać nowe umiejętności. Każdy projekt zostanie oceniony podwójnie – pod względem technicznym na zajęciach informatyki oraz pod względem merytorycznym na przedmiocie, którego dotyczy.

1

Edytor graficzny

Paint. Malowanie kreską, krzywą i kształtami.

Rysowanie linii i krzywych.

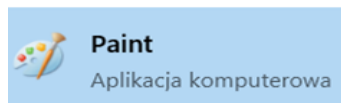
Na tej lekcji nauczysz się:

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

Uruchamianie programu

Najprostszym sposobem na włączenie dowolnego programu w Windows 10 jest znalezienie go w Menu Start:

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

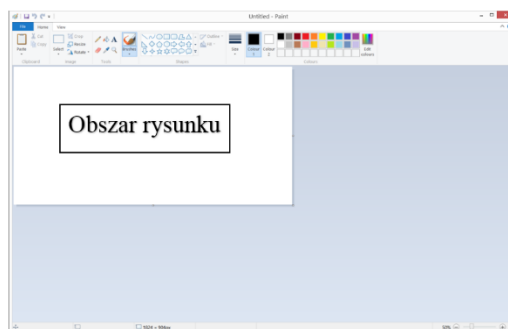


Jeśli na pulpicie utworzony jest skrót do programu, kliknij w jego ikonkę.

Pusty projekt

Po uruchomieniu aplikacji pojawi się okno programu.

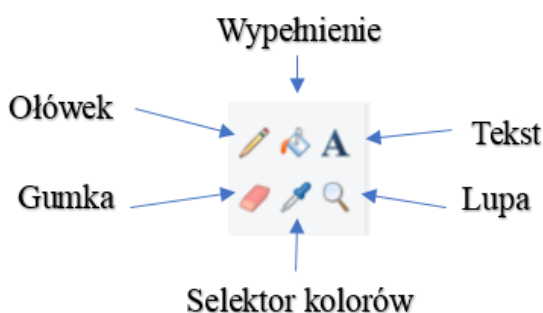
Okno programu składa się z dwóch części – wstążki (znajdują się na niej dwie karty – Narzędzia główne oraz Widok) i obszaru rysunku.



- Wymiary rysunku możesz zmienić poprzez kliknięcie myszą na jeden z kwadracików znajdujących się na brzegu obszaru rysunku (białe pole).
- Aby zmniejszyć wymiary obrazka, należy przytrzymać wciśnięty lewy przycisk myszy i przeciągnąć do wewnątrz.
- 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 oraz Kolory. Narzędzia umożliwią Ci tworzenie i edycje rysunków na ekranie komputera. Oto podstawowe narzędzia:



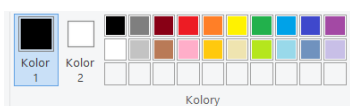
Ołówek



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 zmienia się korzystając z palety kolorów znajdujących 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



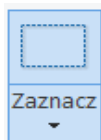
Wypełnienie kolorem służy do wypełniania całych rysunków lub pojedynczych kształtów wybranym kolorem. Kolor wypełnienia wybiera się w ten sam sposób co kolor ołówka.

Gumka



Gumka służy do wymazania części obrazu. Za pomocą gumki możesz zetrzeć wszystko, co narysowałeś. 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.

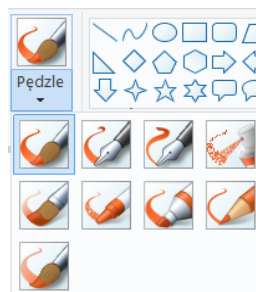


Po wybraniu opcji *Zaznacz* wciśnij i przytrzymaj lewy przycisk myszy. Ruszając myszą, utworzysz prostokąt wokół obszaru, który chcesz zaznaczyć.

Czynność	Skutek
Wciśnięcie klawisza Delete	Usunięcie obszaru zaznaczenia
Przesunięcie myszą do innej części rysunku	Przesunięcie zaznaczenia do wskazanego obszaru
Wciśnięcie klawiszy Ctrl + C	Skopiowanie obszaru zaznaczenia

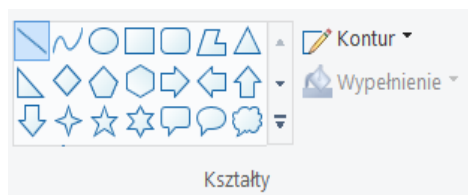
Pędzle

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 kartę **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

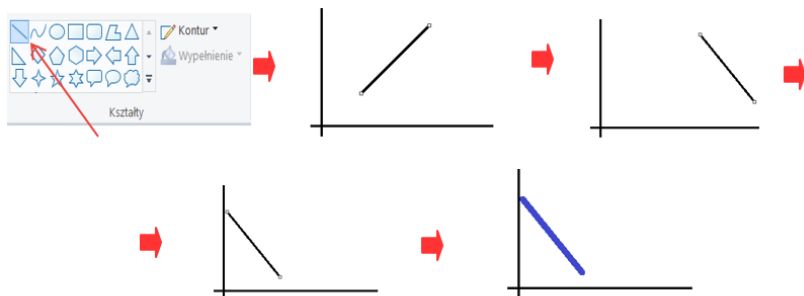
Aby narysować odcinek za pomocą narzędzia Linia, należy wybrać w górnym panelu programu symbol , a następnie nacisnąć lewy przycisk myszy i przeciągnąć po obszarze rysunku.

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

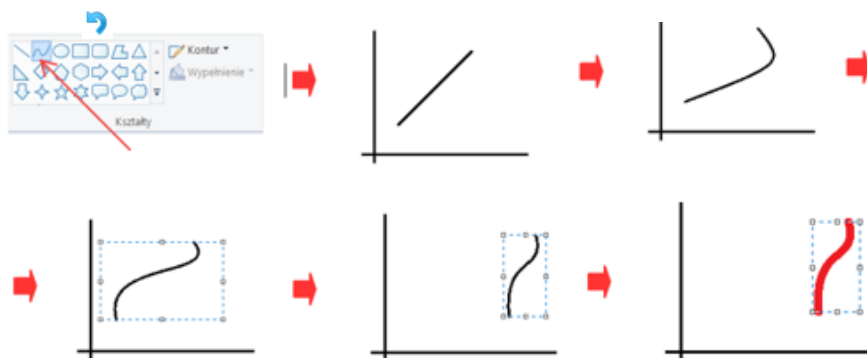
Aby narysować krzywą za pomocą narzędzia Krzywa, należy wybrać w górnym panelu programu symbol , a następnie nacisnąć lewy przycisk myszy i przeciągnąć po obszarze rysunku.

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:

1. Przeniesienie stworzonej krzywej poprzez przytrzymanie kursorem w dowolnym punkcie wewnątrz ramki i poruszanie myszą;
2. Rozciągnięcie lub zwinięcie ramki wraz z krzywą przy użyciu kwadratów w rogach i na bokach ramki,
3. Zmiana koloru i grubości krzywej odbywa się podobnie jak w przypadku zmiany 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 błąd i chcesz wrócić do poprzedniego stanu 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 dyskiety . 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 poniższy rysunek:



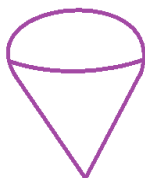
Zapisz projekt w swoim folderze. Plik nazwij „Świąteczna_choinka_imię_nazwisko_klasa”.

Zadanie 2.

Za pomocą **Linii** i **Krzywej**, narysuj:

1. literę A w kolorze czerwonym za pomocą linii,
2. literę C w kolorze niebieskim za pomocą krzywej,
3. lód w kolorze fioletowym za pomocą linii i krzywej,
4. 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:

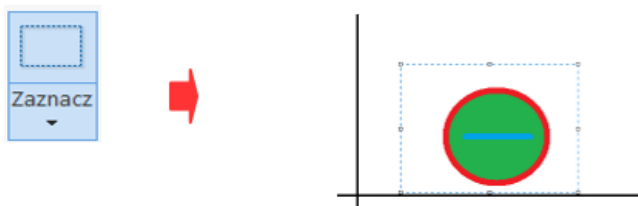
1. kopiować, przenosić i wycinać fragmenty rysunków;
2. Korzystać z gotowych kształtów;
3. dodawać pole tekstowe i z niego korzystać;
4. przenosić rysunek na inny komputer.

Narzędzie wycinania c.d.

Z narzędziem wycinania poznaliśmy się już w poprzednim rozdziale. W tym rozdziale przyjrzymy się mu trochę bliżej.

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

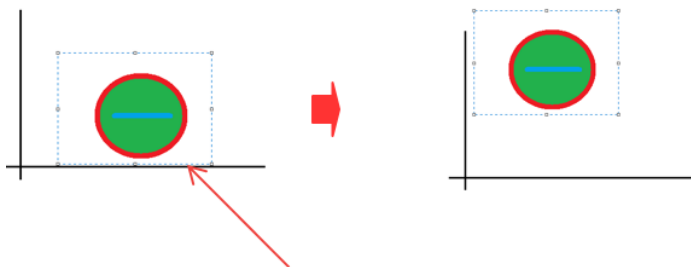
Pojawi się ramka odpowiadająca zaznaczonemu obszarowi:



Zaznaczenie obszaru umożliwia szereg czynności.

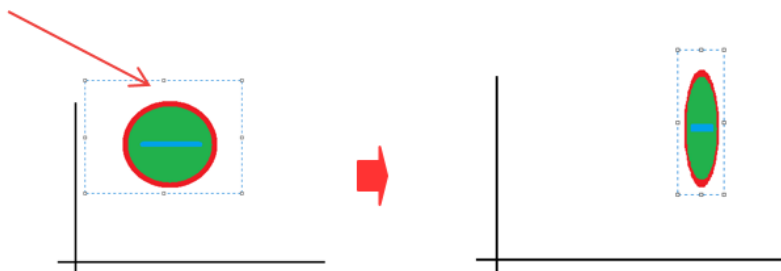
Przeniesienie zaznaczonego fragmentu

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

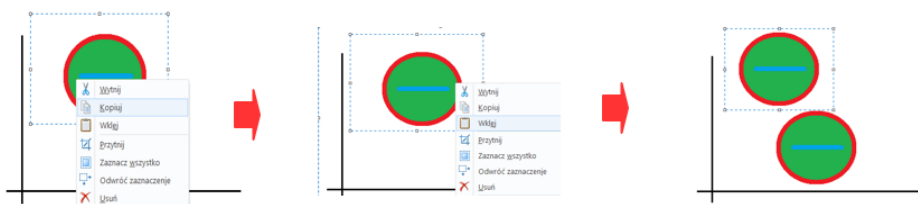
Aby usunąć wybrany obszar, należy wcisnąć klawisz **Delete**.

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.



Czynność

Skutek

Wciśnięcie klawisza **Delete**

Usunięcie obszaru zaznaczenia

Przesunięcie myszą do innej części rysunku

Przesunięcie zaznaczenia do wskazanego obszaru

Kształty geometryczne

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

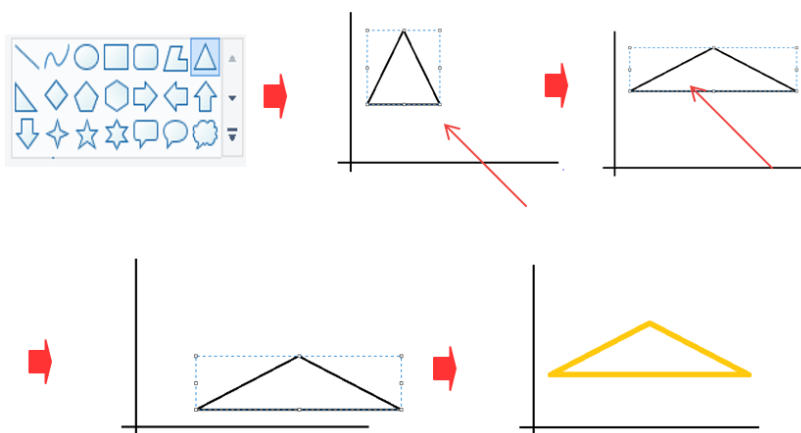
- okrąg
- prostokąt
- trójkąt

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:

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

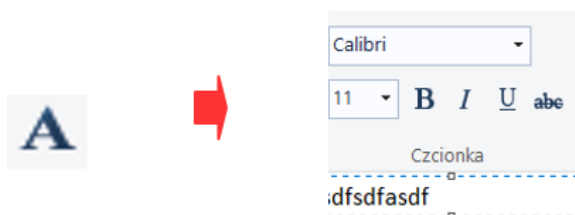
Gdy uznasz, że figura jest już gotowa, kliknij lewym przyciskiem myszy na dowolny punkt poza obszarem obramowania.



Pole tekstowe

Narzędzie to służy do dodawania tekstu do rysunku. Aby z niego skorzystać, wybierz narzędzie Pola Tekstowego  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, m.in. 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 zanieść do domu rysunek zapisany w pliku na komputerze szkolnym, możesz go umieścić na dysku USB.

Aby zapisać rysunek na dysku USB wykonaj następujące kroki:

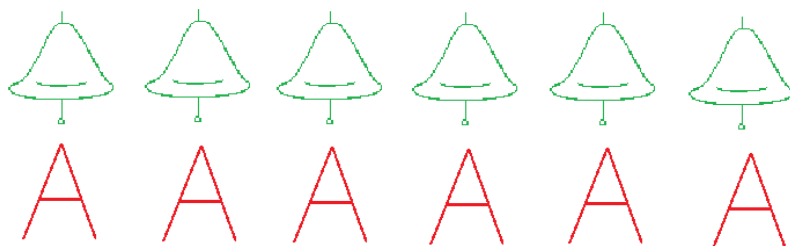
1. Wejdź w **Mój komputer** i znajdź dysk USB.
2. Znajdź plik, który chcesz skopiować i przenieś go na USB korzystając z opcji **Kopiuj** i **Wklej**.
3. Upewnij się, że skopiowany plik znajduje się na USB.
4. Bezpiecznie wyśnij dysk przenośny.

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.

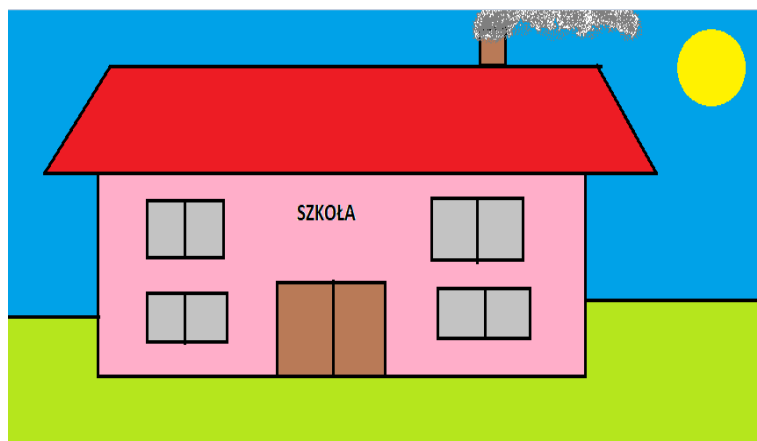
Następnie korzystając z tego samego narzędzia, stwórz pod istniejącymi obrazkami dwa rzędy identycznych, narysowanych przez siebie na poprzednich zajęciach figur, kopiując te istniejące i przenosząc je. Pamiętaj, że narzędzie **Wycinania** umożliwia zmianę wielkości stworzonego kształtu. Oto przykładowy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



Zadanie 2.

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

Za przykład może Ci posłużyć poniższy rysunek:



Zapisz stworzony rysunek w swoim folderze. Plik nazwij „Moja_Szkoła_Imię_Nazwisko_Klasa”.

Pora, by zastosować poznane umiejętności. Poniżej znajdziesz propozycje projektów łączących informatykę i zajęcia plastyki. Zrealizuj jeden z nich lub inny, wyznaczony przez Twojego nauczyciela.

Projekt międzyprzedmiotowy

Edytor graficzny

Informatyka / plastyka. „Góry”

Projekt 1. Góry

Korzystając z poznanych narzędzi i funkcji programu Paint, stwórz rysunek pt.: „Góry”.

Przy tworzeniu rysunku wykorzystaj:

1. linię i krzywą,
2. gotowe kształty geometryczne,
3. wypełnienie kolorem,
4. ołówek i przynajmniej jeden z pędzli,
5. narzędzie wycinania,
6. 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 „Mozaiki”

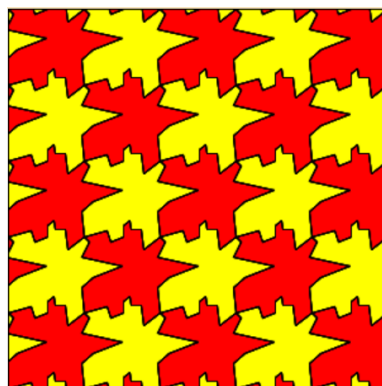
Projekt 2. Mozaiki

Korzystając z poznanych narzędzi i funkcji programu Paint, stwórz własną mozaikę. Inspiracją może być kalejdoskop lub mozaiki Eschera. Wyszukaj w sieci więcej informacji na ich temat.

Przy tworzeniu rysunku wykorzystaj:

1. linię i krzywą,
2. gotowe kształty geometryczne,
3. wypełnienie kolorem,
4. narzędzie wycinania,
5. 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 tekstowy

LibreOffice Writer / MS Word

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 edytora tekstu jest program Notatnik, który znajdziesz w akcesoriach systemu.

W tym rozdziale przedstawiamy podstawowe działanie dwóch edytorów tekstu:



LibreOffice Writer oraz



Microsoft Word

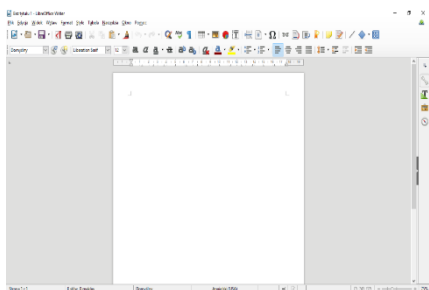
Uruchamianie programu

Jeśli program jest już zainstalowany na Twoim komputerze, to znajdziesz go w Menu Start poprzez wpisanie jego nazwy w polu wyszukiwania. Jeśli na pulpicie został utworzony skrót, wystarczy na niego kliknąć.

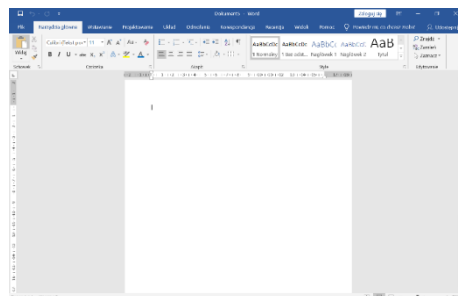
Pusty dokument

Po uruchomieniu programu pojawi się pusty dokument tekstowy. Białe pole to karta dokumentu, czyli miejsce służące do pisania tekstu.

LibreOffice Writer



Microsoft Word



Litery diakrytyczne

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

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

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

a lewy Alt + a

† lewy Alt + I

- ś lewy Alt + s

ć lewy Alt + c

ń lewy Alt + n

z lewy Alt + z

e lewy Alt + e

ó lewy Alt + o

- Ż lewy Alt + x

Wielkie litery

Wielkie litery tworzy się poprzez wciśnięcie klawisza **Shift** wraz z literą na klawiaturze.

Na przykład:

A Shift + a

B Shift + b

C Shift + c

Wielkie „polskie litery” (litery diakrytyczne)

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

Znaki interpunkcyjne

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

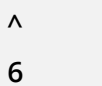
kropka	.	wielokropek	...
przecinek	,	znak zapytania	?
średnik	;	wykrzyknik	!
dwukropek	:	nawias	()
myślnik	-	cudzysłów	„”

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 :, 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	przejsięcie do następnej linii, potwierdzenie operacji	Backspace	usunięcie ostatniego znaku
Spacja	odstęp w linii między znakami	Esc	anulowanie operacji
Alt	polskie litery, kombinacje klawiszowe	Ctrl	kombinacje klawiszowe
Caps Lock	ciągłe pisanie wielkimi literami	Tab	duży odstęp w linii tekstu
Shift	wielkie litery, znaki interpunkcyjne, kombinacje klawiszowe		

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 życie nie tylko podczas pracy z edytorem tekstu.

Zadanie 1.

Otwórz nowy dokument.

Przepisz poniższe słowa, korzystając z polskich znaków diakrytycznych:

łążącą, łożącą, łożmyż, łożże, łżącą, żąłeś, żęłaś, żółcą, żółcę, żółć.

Zadanie 2.

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

Karolina krzyknęła: Adam, zawołaj Laure!

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 – wkleja, 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:

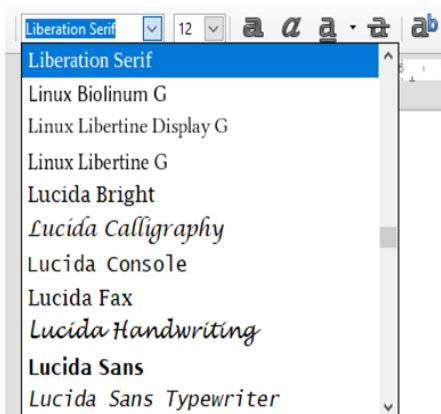
1. pisać i edytować różnorodny tekst;
2. podstawowych zasad pisania i edycji tekstu;
3. tworzyć jednopoziomową listę numerowaną.

Pisanie i edycja tekstu

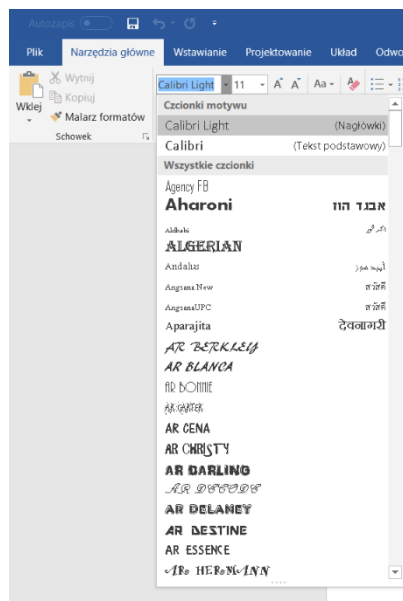
Edytor tekstowy pozwala na wybór i zmianę wielu cech tekstu, wpływających na jego wygląd.

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



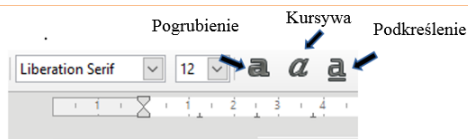
Wybierając rodzaj i krój czcionki, możesz zmienić wygląd tekstu.

Przykład

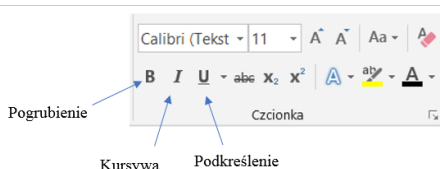
Najdłuższy dzień roku. (Times New Roman, 13)
Najkrótszy dzień roku. (**Comic Sans MS**,15)

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

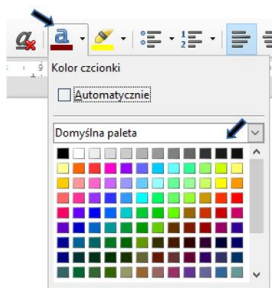


Przykład

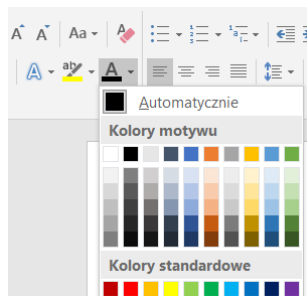
Najdłuższy dzień roku. (Liberatrion Serif, 12)
Najkrótszy **dzień** roku. (Liberatrion Serif, 12)

Zmianę 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 powyłamywanymi nogami. W czasie suszy szosa sucha.
(Agency FB, 12)

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

Zasady pisania i edycji tekstu

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

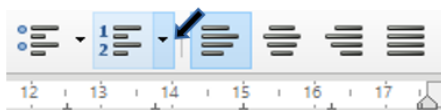
Poniżej przedstawione są najważniejsze zasady poprawnej pisowni, do których należy się stosować:

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. Nie stawiaj spacji przed i po tekście ujętym we wnętrzu nawiasu lub cudzysłowu.
7. Pamiętaj o zasadach ortografii i interpunkcji.

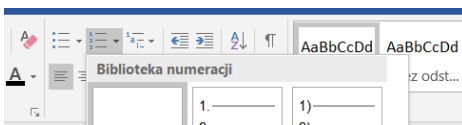
Listy jednopoziomowe

Niektóre informacje, jak na przykład plan wycieczki szkolnej czy lista zakupów, wygodnie jest zapisać w postaci list numerowanych. Na karcie **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 wolny.
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. (Liberation Serif, 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.: „Niezwyczajny dzień w szkole”, inspirowany książką J.J. Sempego i R. Goscinnego „Mikołajek”.

Musi on zawierać:

1. tytuł napisany czcionką Times New Roman, 16, pogrubioną, kolorem niebieskim
2. listę numerowaną, napisaną czcionką Times New Roman, 12, kolorem czarnym

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

Formatowanie tekstu.

Akapity, wcięcia, interlinie.

Na tej lekcji nauczysz się:

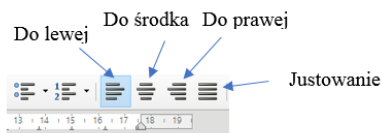
1. wyrównywać tekst na stronie;
2. dzielić tekst na akapity, w tym dodawać wcięcia w tekście oraz odstępy między liniami;
3. formatowania tekstu;
4. praktycznego zastosowania nabytych umiejętności.

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



Wyrównanie do lewej



Przechoń spokojnie przez hałas i pośpiech i pamiętaj, jaki spokój można znaleźć w ciszy. O ile to możliwe, bez wyrzekania się siebie, bądź na dobrej stopie ze wszystkimi.

Wyrównanie do środka



Wypowiadaj swą prawdę jasno i spokojnie i wysłuchaj innych,
nawet tępych i nieświadomych; oni też mają swoją opowieść.
Unikaj głośnych i napastliwych - są udręką ducha.

Wyrównanie do prawej



Porównując się z innymi, możesz stać się próżny lub zgorzkniały,
zawsze bowiem znajdziesz gorszych i lepszych od siebie.
Niech twoje osiągnięcia, zarówno jak i plany,
będą dla ciebie źródłem radości.

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

Justowanie



Wykonuj swą pracę z sercem, jakkolwiek byłaby skromna; ją jedynie posiadasz w zmiennych kolejach losu. Bądź ostrożny w interesach, na świecie bowiem pełno oszustwa, niech to ci jednak nie przysłoni prawdziwej cnoty. Wielu ludzi dąży do wzniosłych ideałów i wszędzie życie jest pełne heroizmu.

[fragmenty Desideraty, Max Ehrmann, 1927]

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.

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 Księżyca 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 prawej: dla zapisu dat,
 - do środka: dla tytułów, nagłówków,
 - justowanie: dla zasadniczego tekstu.

Zadanie 1.

Otwórz stworzony na poprzednich zajęciach dokument tekstowy.
Przepisz poniższy tekst w dokładnie takiej samej formie. (Times New Roman, 12)

Jan Brzechwa

Atrament i kreda (fragment)

Wzdychała kreda: "Wciąż jestem biała,

Nie chcę być biała!..." No i-szczerniała.

Jęczał atrament: "O, losie marny,

Wciąż jestem czarny, kompletnie czarny,

Jak gdyby we mnie kto smołę przelał.

Nie chcę być czarny! Dość już!" I zbiełał.

Zadanie 2.

Przepisz poniższy tekst, zachowując podział na akapity.
Zastosuj justowanie. (Liberatrion Sans, 12)

W 1030 Mieszko, zabezpieczony sojuszem z Węgrami, raz jeszcze najechał Saksonię. Tymczasem jego południowy sprzymierzeniec zaatakował Bawarię, przejściowo zajmując Wiedeń.

W odpowiedzi cesarz zorganizował kolejną wyprawę przeciw królowi Polski, tym razem organizując szerszą koalicję antymieszkowską.

Pora, by zastosować poznane umiejętności. Poniżej znajdziesz propozycje projektów łączących informatykę i język polski. Zrealizuj jeden z nich lub inny wyznaczony przez Twojego nauczyciela.

Projekt międzyprzedmiotowy

Edytor tekstu

Informatyka / język polski.

Opowiadanie „Niezwyczajny dzień”

Na podstawie stworzonego przez siebie planu napisz opowiadanie do 1500 znaków. Poniżej wyszczególnione są elementy, które powinno zawierać opowiadanie.

1. tytuł:
 - wyśrodkowany,
 - czcionką Times New Roman 15,
 - kolorem niebieskim,
 - pogrubiony,
2. co najmniej 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 12,
4. twój podpis i datę, napisane kursywą i wyrównane do prawej, umieść na końcu opowiadania.

Projekt międzyprzedmiotowy

Edytor tekstu

Informatyka / język polski

Opowiadanie „Niesamowita historia”

Wyobraź sobie, że przeniosłeś/ przeniosłaś się w czasie i jesteś świadkiem ważnego wydarzenia z historii Polski. Opisz to, co zobaczyłeś/ zobaczyłaś.
(II)

Stwórz plan swojego opowiadania. Na jego podstawie napisz opowiadanie do 1500 znaków. Poniżej wymienione są elementy, które powinno zawierać opowiadanie:

5. tytuł:
 - wyśrodkowany,
 - czcionką Times New Roman 15,
 - kolorem niebieskim,
 - pogrubiony,
6. co najmniej 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,
7. tekst napisany czcionką Times New Roman 12,
8. twój podpis i datę, napisane kursywą i wyrównane do prawej, umieść na końcu opowiadania.

3

Prezentacje

LibreOffice Impress / MS Powerpoint

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 one mają podobne funkcje. W tym rozdziale przedstawiamy podstawowe działania dwóch z nich:

LibreOffice Impress

oraz

Microsoft PowerPoint



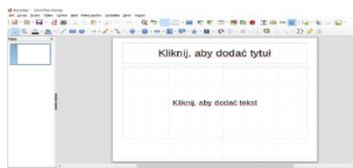
Uruchamianie programu

Jeśli program jest już zainstalowany na Twoim komputerze, to znajdziesz go w Menu Start poprzez wpisanie jego nazwy w polu wyszukiwania. Jeśli na pulpicie został utworzony skrót, wystarczy na niego kliknąć.

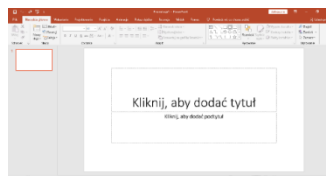
Pusty projekt

Po uruchomieniu programu pojawi się pusty projekt prezentacji.

LibreOffice Impress



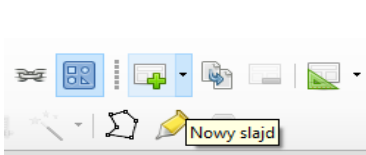
Microsoft PowerPoint



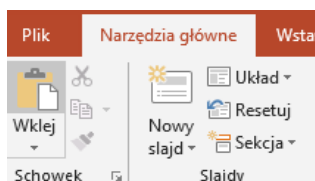
Dodawanie slajdów

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

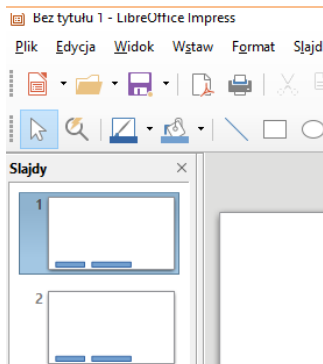


Microsoft PowerPoint

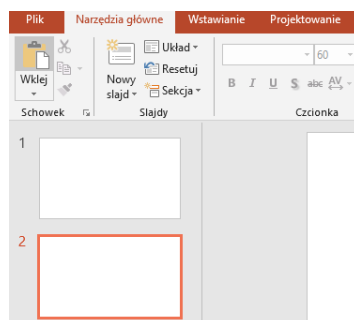


Nowe slajdy będą pojawiać się w zakładce po lewej stronie ekranu. Między różnymi slajdami przechodzi się klikając na miniaturkę wybranego slajdu na liście.

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

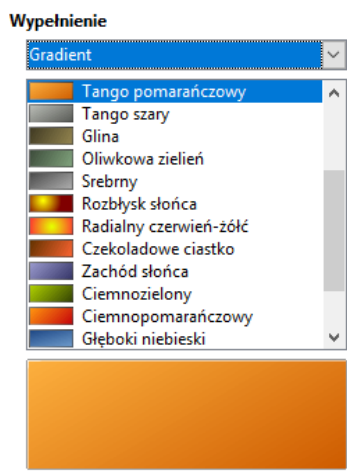
Dobrze dobrany kolor tła slajdów sprawi, że Twoja prezentacja będzie wyglądała atrakcyjnie.

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.

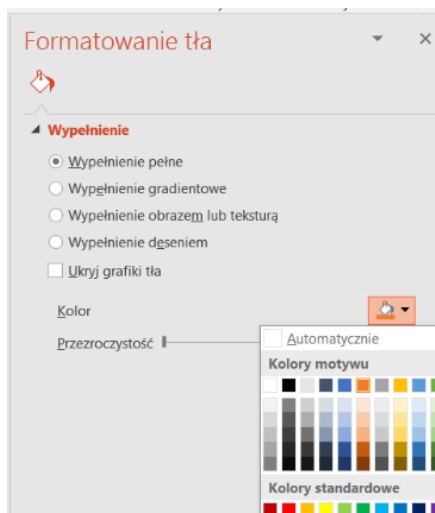
Przykładowo wybierz rodzaj wypełnienia **Gradient** i kolor **Tango pomarańczowy**.



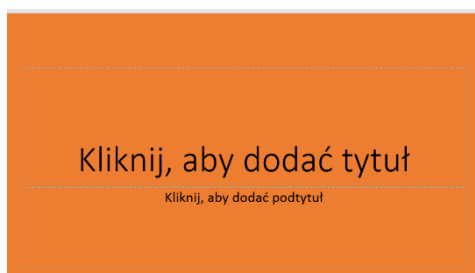
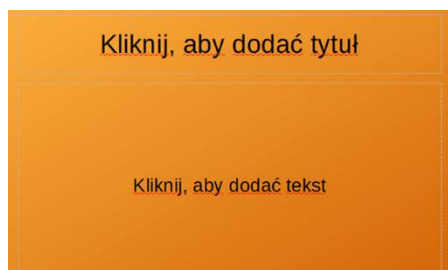
Microsoft PowerPoint

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, poniżej zaproponowane 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.

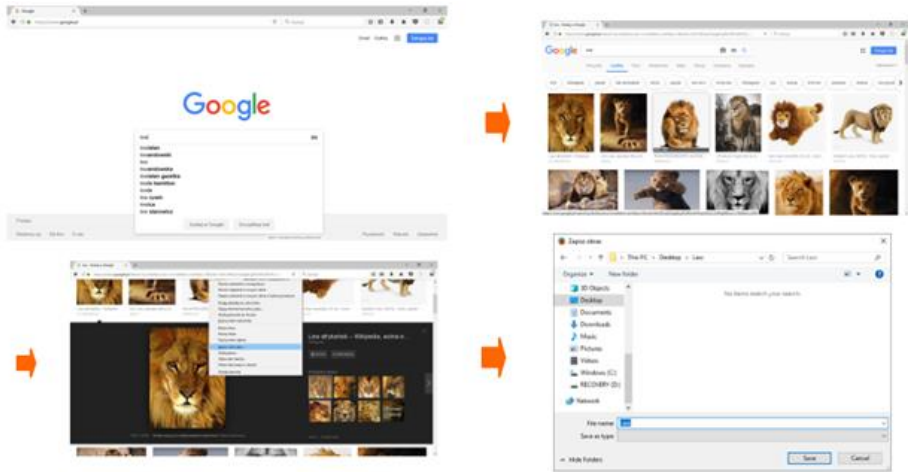
Dodawanie ilustracji

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 być ostrożnym, 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).

Aby wyszukać i zapisać zdjęcia na dysku komputera:

1. uruchom wybraną przeglądarkę (np. Edge czy Mozilla Firefox),
2. wejdź na stronę wyszukiwarki (np. google.com),
3. wpisz w polu wyszukiwania szukane hasło (np. „lew”),
4. przejdź do zakładki Grafika,
5. kliknij prawym przyciskiem myszy na wybrane zdjęcie,
6. ponownie kliknij na obrazek lewym przyciskiem myszy i wybierz opcję **Zapisz obraz jako**,
7. wybierz folder, do którego chcesz zapisać zdjęcie . Nadaj nazwę dla zapisywanemu plikowi i kliknij opcję **Zapisz**.

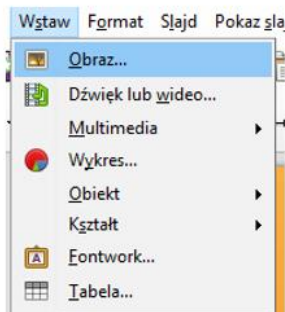
Cała procedura może wyglądać następująco:



Dodajmy do prezentacji ilustrację.

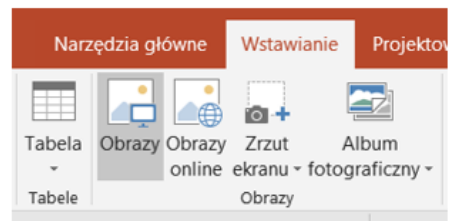
LibreOffice Impress

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.

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 poniższe zasady.

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 przykładowy slajd:



- Lew afrykański to mięsożerny gatunek dużego ssaka lądowego z rodziny kotowatych
- Drugi po tygrysie największy wielki kot na świecie, może ważyć 150-200 kg, największy Lew w historii ważył 375 kg
- Obecnie na wolności występuje jedynie w Afryce, kiedyś zamieszkiwał też między innymi Europę i Azję
- W ciągu ostatnich 25 lat jego liczebność zmalała czterokrotnie, głównie przez działalność człowieka
- Lwy jako jedyne koty żyją w zorganizowanych grupach społecznych

Zastanów się chwilę, czy slajd ten został stworzony zgodnie z zasadami dobrej prezentacji.

Oczywiście jest to przykład niepoprawnie stworzonego slajdu.

Oto kilka błędów, które popełnił twórca:

1. brak tytułu slajdu;
2. mało czytelny tekst (za mała czcionka, kolor czcionki źle dobrany do koloru tła);
3. tekst jest napisany pełnymi zdaniami, zamiast krótkimi hasłami;
4. nieczytelne ilustracje (małe rozmiary zdjęć, zbyt duża liczba zdjęć).

Zastanów się, jak można poprawić ten slajd.



Lew afrykański

- Mięsożerny ssak z rodzaju kotowatych
- Na wolności jedynie w Afryce
- Jego liczebność dramatycznie spada
- Jedyne koty żyjące w stadach

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.

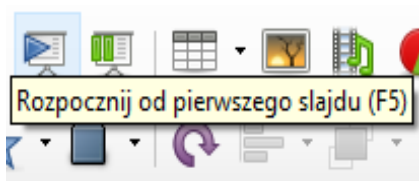
1. Prezentuj z pamięci. Nie czytaj tekstu z prezentacji. Twoja prezentacja jest tłem i ilustracją twojej wypowiedzi, nie jej treścią.
2. Bądź zwrócony twarzą do klasy. Stój prosto. Mów do koleżanek i kolegów, starając się zmieniać osoby, na które patrzysz.
3. Zwróć uwagę na swój głos. Nie możesz mówić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

Tryb prezentacji

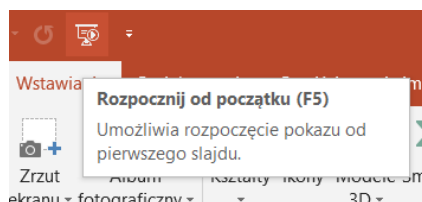
Tryb prezentacji to wygodny sposób na przedstawienie gotowej prezentacji. W tym trybie prezentacja wyświetlana jest slajd po slajdzie.

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. wejdź na stronę:
<https://www.nasa.gov/sites/default/files/thumbnails/image/abifulldiscindex.jpg>
i zapisz na dysku zdjęcie Ziemi, a następnie dodaj je do tytułowego slajdu;
2. wejdź na stronę:
<https://www.nasa.gov/sites/default/files/thumbnails/image/iss056e162811.jpg>
i zapisz na dysku zdjęcie Ziemi, a następnie dodaj je do drugiego slajdu;
3. wejdź na stronę:
https://www.esa.int/spaceinimages/Images/2013/12/Europe_in_the_Solar_System2
i zapisz na dysku grafikę obrazującą Układ Słoneczny, a następnie dodaj ją do trzeciego slajdu.

Zapisz prezentację.

Pora, by zastosować poznane umiejętności. Poniżej znajdziecie propozycje projektu grupowego łączącego informatykę i historię. Zrealizujcie ten projekt lub inny wyznaczony przez nauczyciela.

Projekt międzyprzedmiotowy

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 okazja, by tego doświadczyć. Ruszacie w czas przeszły.

Dobierzcie się w grupy trzyosobowe. Ustalcie między sobą rok, w który się przenosicie. Sprawdźcie w dostępnych źródłach, kto władał Polską w tym czasie. Każda z grup niech stworzy prezentację multimedialną na temat wybranego władcy oraz przygotuje 5 - minutowe wystąpienie na jej podstawie.

Rozdzielcie zadania tak, żeby każdy z Was miał 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.

4

Arkusz kalkulacyjny LibreOffice Calc./ MS Excel

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. 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:

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

Aplikacje

Arkusz kalkulacyjny to aplikacja służąca do operacji na liczbach. Dostępne są różne arkusze kalkulacyjne.

W tym rozdziale przedstawiamy podstawowe działanie dwóch arkuszy kalkulacyjnych:

LibreOffice Calc



oraz

Microsoft Excel



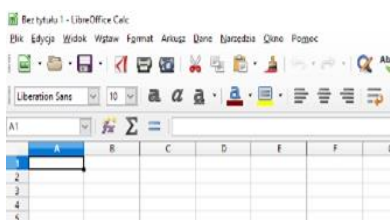
Uruchamianie programu

Jeśli program jest już zainstalowany na Twoim komputerze, to znajdziesz go w Menu Start poprzez wpisanie jego nazwy w polu wyszukiwania. Jeśli na pulpicie został utworzony skrót, wystarczy na niego kliknąć.

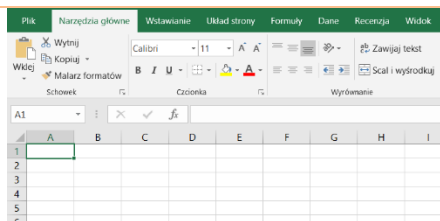
Nowy arkusz

Po uruchomieniu programu otworzy się nowe okno z pustym arkuszem.

LibreOffice Calc

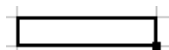


Microsoft Excel



Komórki, kolumny, wiersze

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

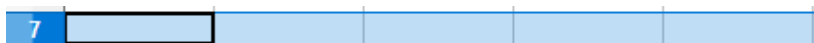


Kolumna to pionowy zbiór komórek znajdujących się “w jednej linii”.



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.

Formatowanie komórek, kolumn i wierszy

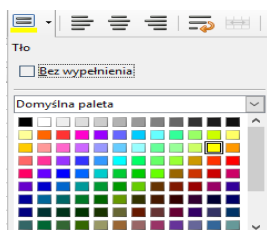
Arkusz kalkulacyjny umożliwia zmianę tła, obramowania oraz czcionki pojedynczej komórki lub kilku zaznaczonych komórek na raz.

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

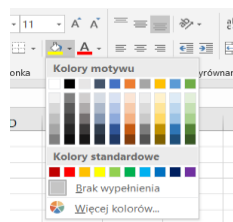
Zmiana tła

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



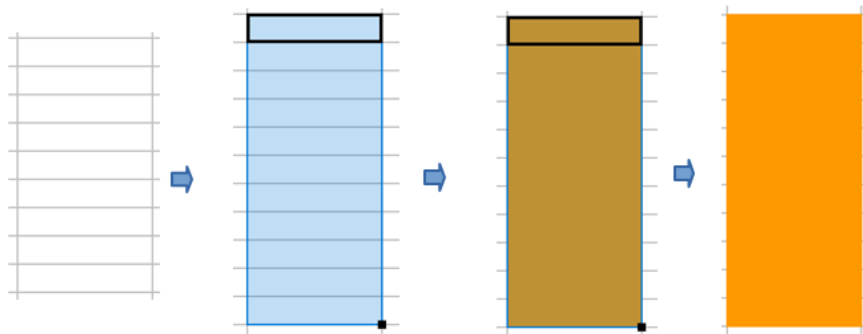
Efekt zmiany koloru komórki:



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 nas interesuje.

Efekt zmiany koloru zbioru komórek:

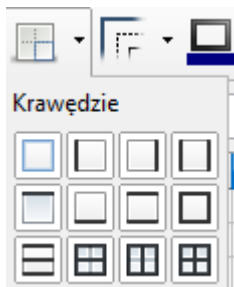


Obramowanie

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

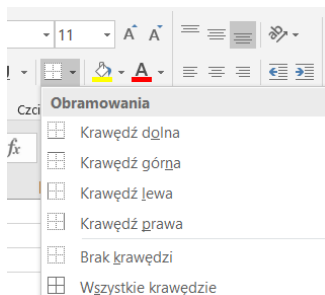
LibreOffice Calc

Wybierz narzędzie **Krawędzie** znajdujące się w górnym pasku.

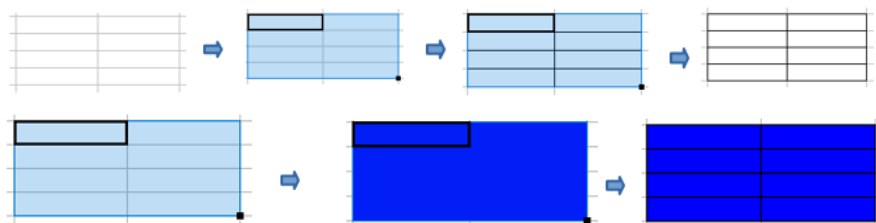


Microsoft Excel

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



Wybierz opcję **wszystkie krawędzie**. Cały proces wygląda następująco:



Zapisywanie pliku

Dobłą 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 dyskiety . 
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 dyskiety, aby zapisać postępy w pracy.

Zadanie 1.

Podaj adresy (literę kolumny i numer wiersza) pokolorowanych komórek:

- żółta:,
- niebieska:,
- czerwona:,
- zielona:

	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:

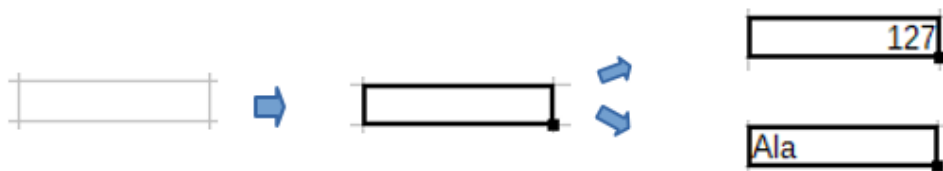
1. wpisywać wartości do komórek w arkuszu;
2. dodać lub odjąć wartości dwóch komórek;
3. mnożyć lub dzielić wartości dwóch komórek;
4. usunąć zawartość wypełnionej komórki;
5. czym jest pasek formuły i jak z niego korzystać.

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:

A1a

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.

Program traktuje przecinek jako część liczby. Wszystkie znaki inne niż cyfry traktowane są jak tekst.

Operacje matematyczne

Jak zostało to 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			

2. Wpisz do zaznaczonej komórki znak równości "=":

	A	B	C
1	5	12	=
2			

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
2			

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+
2			

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
2			

 ➡

	A	B	C
1	5	12	17
2			

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

Odejmowanie

Przy odejmowaniu, kroki 1, 2 oraz 3 są dokładnie takie same, jak w przypadku dodawania. Jedyna różnica 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 ów” zamiast „Kraków”, możemy łatwo poprawić ten błąd. Używając klawisza **Backspace** 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”.