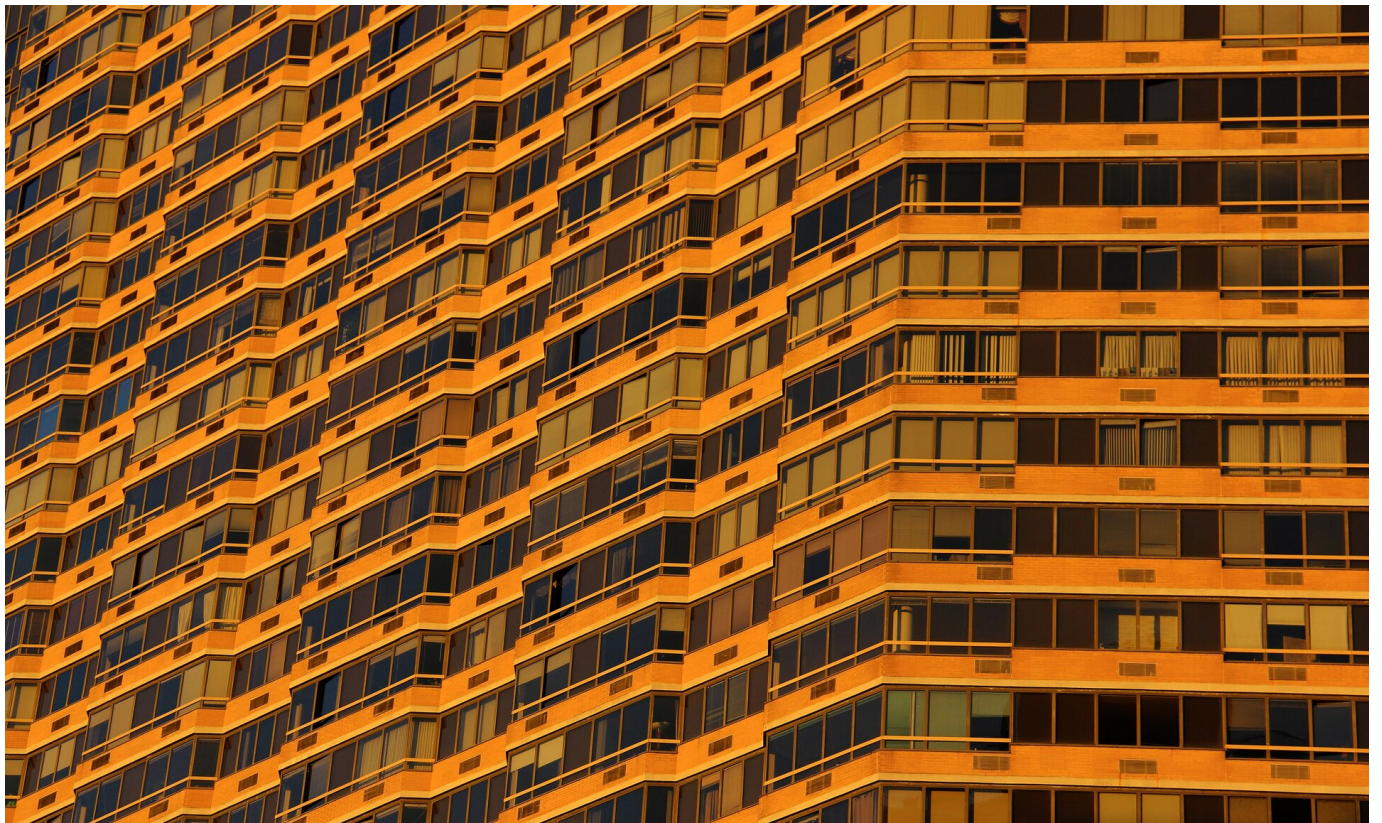




Algorytmy iteracyjne – obliczanie silni

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Gra edukacyjna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Algorytmy iteracyjne – obliczanie silni

Źródło: Alan Weiner, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Iteracja polega na powtarzaniu tej samej operacji w pętli ściśle określoną liczbę razy lub aż do spełnienia założonego warunku. Algorytmy, które powtarzają dane czynności, aby osiągnąć zamierzony cel, nazywa się algorytmami iteracyjnymi.

W tym e-materiale omówimy zapisywanie takich algorytmów w postaci schematów blokowych.

Więcej informacji o algorytmach iteracyjnych znajdziesz w e-materiałach:

- [Algorytmy iteracyjne](#),
- [Algorytmy liczbowe](#),
- [Algorytmy iteracyjne i liczbowe – potęgowanie liczb](#).

Więcej zadań? Sięgnij do e-materiału [Algorytmy iteracyjne i liczbowe – zadania maturalne](#).

Twoje cele

- Przeanalizujesz problem i przedstawisz jego rozwiązanie za pomocą schematu blokowego.
- Wyjaśnisz działanie schematów blokowych na konkretnych przykładach.

- Skonstruujesz własny algorytm iteracyjny, który będzie obliczał siłnię.

Przeczytaj

Jak działa algorytm iteracyjny?

Mechanizm algorytmów [iteracyjnych](#) jest oparty na powtarzaniu pewnych operacji. Odbywa się to na dwa podstawowe sposoby:

1. powtarzanie operacji z góry określoną liczbę razy,
2. powtarzanie operacji do momentu spełnienia pewnego warunku.

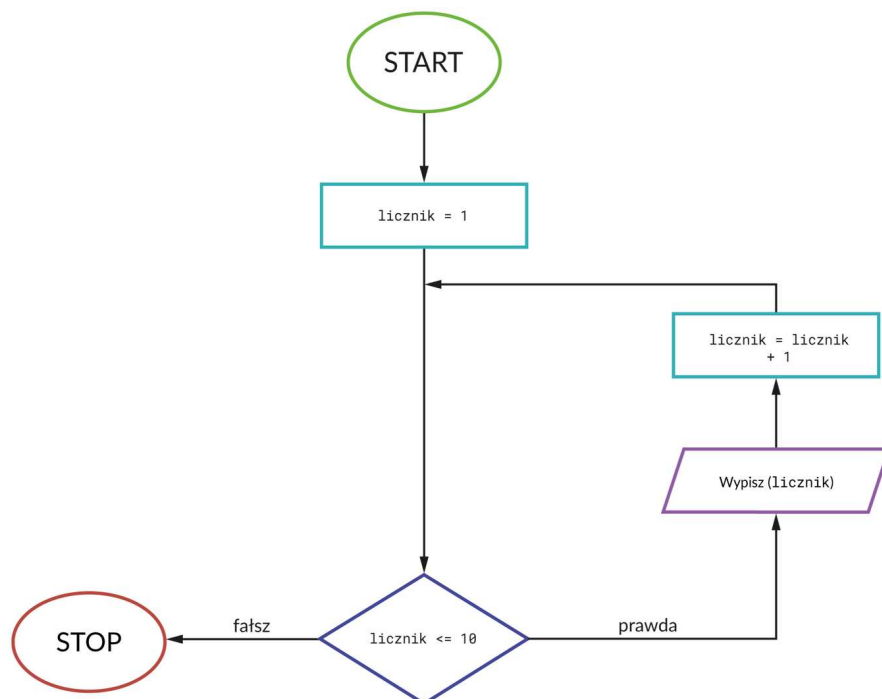
Oto przykładowe zastosowania algorytmów iteracyjnych:

- wypisanie liczb o określonych właściwościach z przedziału $\langle a, b \rangle$, gdzie wartości a i b podaje użytkownik,
- wypisanie podanego tekstu dokładnie n razy, gdzie n podaje użytkownik,
- zmniejszanie ilości pozostałego czasu, aż do jego wyczerpania,
- powtarzanie prośby o udzielenie odpowiedzi do momentu, w którym użytkownik poda odpowiedź poprawną.

Prezentowanie algorytmu iteracyjnego za pomocą schematu blokowego

Jak wygląda graficzny zapis iteracji w algorytmie? Posłużymy się przykładem:

Przyjmijmy, że chcemy wypisać wszystkie liczby naturalne z przedziału od 1 do 10 włącznie.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Ważne!

W schematach blokowych wszelkie operacje na **zmiennych** zapisywane są w blokach o kształcie prostokątów.

Przeanalizujmy ten algorytm:

1. Ustawiamy wartość zmiennej **licznik** na 1.
2. Sprawdzamy, czy wartość zmiennej **licznik** jest mniejsza bądź równa 10.
3. Jeśli tak, wypisujemy zmienną **licznik**, a następnie zwiększamy jej wartość o 1 i ponownie sprawdzamy warunek z punktu 2.
4. Jeżeli wartość zmiennej **licznik** jest większa od 10, kończymy wykonywanie algorytmu.

Przykład: tworzenie algorytmu iteracyjnego krok po kroku

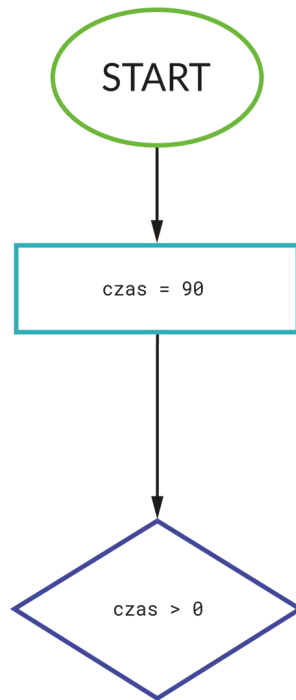
Napiszmy algorytm, który pozwoli sprawdzić, ile czasu zostało studentom do zakończenia egzaminu. Zakładamy, że egzamin trwa 90 minut.

1. Najpierw należy przypisać zmiennej **czas** wartość początkową – czyli 90 (liczymy czas w minutach).



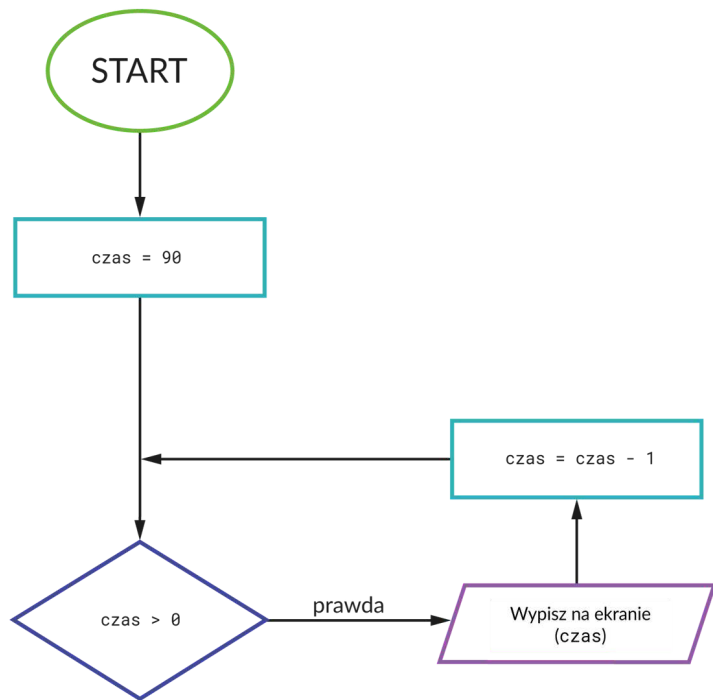
Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

2. Sprawdzamy, czy wartość zmiennej `czas` jest większa niż 0.



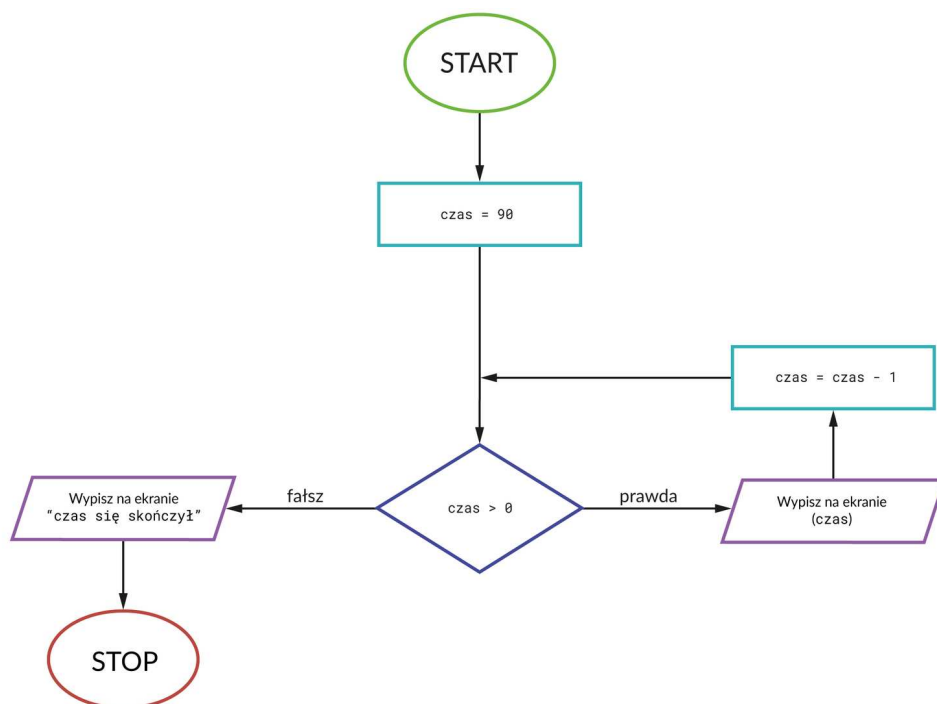
Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

3. Jeśli tak, wyświetlamy informację, ile czasu pozostało studentom. Następnie zmniejszamy wartość zmiennej `czas` o 1 i ponownie sprawdzamy, czy jest ona nadal większa od 0.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

4. Jeśli czas już upłynął (wartość zmiennej czas jest równa 0), wypisujemy komunikat o zakończeniu egzaminu oraz kończymy pracę z algorytmem.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Słownik

iteracja

powtarzanie tej samej operacji z góry określoną liczbę razy lub do momentu spełnienia określonego warunku

zmienna

element języka programowania mający nazwę, wartość oraz miejsce w pamięci komputera, w którym zapisana jest wartość zmiennej; do zmiennej można odwoływać się za pomocą nazwy

Gra edukacyjna

Definicja: Silnia liczby naturalnej n (zapis symboliczny: $n!$)

Iloczyn kolejnych liczb naturalnych od 1 do n .

Przykład:

$$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$

Ważne!

Wartość $0!$ oraz $1!$ wynosi 1.

Polecenie 1

Przedstaw za pomocą narzędzia Blockly algorytm obliczania wartości silni liczby naturalnej n pobieranej od użytkownika.

Przygotowane rozwiązanie przetestuj dla n równego 4.

Specyfikacja problemu:

Dane:

- n – liczba naturalna z przedziału $\langle 0, 170 \rangle$; wartość, dla której obliczamy silnię

Wynik:

Na standardowym wyjściu wypisana jest silnia liczby n .

Ważne!

Zdefiniowany w specyfikacji przedział wartości liczby naturalnej n wynika z ograniczeń narzędzia Blockly. Jeżeli podana przez użytkownika liczba będzie większa od 170, program jako wartość silni zwróci ciąg znaków „Infinity” oznaczający nieskończoność.

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w prezentacji.

Specyfikacja dla algorytmu:

1

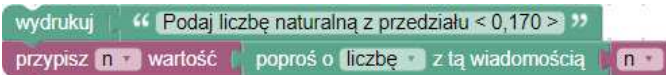
Dane:

- n – liczba naturalna z przedziału $\langle 0, 170 \rangle$; wartość, dla której obliczamy silnię

Wynik:

Na standardowym wyjściu prezentowana jest wartość zmiennej `wynik` równa silni liczby n .

2



Wczytanie wartości zmiennej n

Na początku należy wczytać liczbę, której silnia ma być wyliczona (czyli liczbę n).

3

Teraz zastanówmy się, jak przedstawić silnię w sposób iteracyjny. Zgodnie z definicją, silnia to iloczyn kolejnych liczb naturalnych.

4

Zatem jest to odpowiednia sytuacja, aby użyć iteracji. Przedstawimy silnię jako iloczyn liczb naturalnych, gdzie każda następna jest większa od wcześniejszej o 1.

```

wydrukuj "Podaj liczbę naturalną z przedziału < 0,170 >"
przypisz n wartość poproś o liczbę z tą wiadomością n
przypisz i wartość 1
przypisz wynik wartość 1
  
```

5

Zdefiniowanie zmiennych wynik oraz i

Rozpocznijmy od zdefiniowania zmiennej wynik przechowującej rezultat obliczeń oraz nadania jej wartości 1; będziemy ją mnożyć przez kolejne liczby naturalne $i \leq n$, rozpoczynając od liczby 1. Dokonany przez nas wybór wartości początkowej wyniku z własności:

$$0! = 1$$

6

```

wydrukuj "Podaj liczbę naturalną z przedziału < 0,170 >"
przypisz n wartość poproś o liczbę z tą wiadomością n
przypisz i wartość 1
przypisz wynik wartość 1
powtarzaj dopóki i ≤ n
wykonaj przypisz wynik wartość wynik × i
  
```

Utworzenie pętli

Operację mnożenia wykonywać będziemy, dopóki liczba naturalna i będzie mniejsza lub równa wartości zmiennej n .

```

wydrukuj "Podaj liczbę naturalną z przedziału < 0,170 >"
przypisz n wartość poproś o liczbę z tą wiadomością n
przypisz i wartość 1
przypisz wynik wartość 1
powtarzaj dopóki i ≤ n
wykonaj
  przypisz wynik wartość wynik × i
  przypisz i wartość i + 1
  
```

7

Umieszczenie w pętli instrukcji zwiększania wartości zmiennej i

Pamiętajmy, że skoro chcemy wartość zmiennej wynik mnożyć przez kolejne liczby, to na końcu każdej iteracji zwiększamy wartość zmiennej i o 1.

8

Instrukcję przypisania zmiennej wynik nowej wartości oraz instrukcję zwiększania o 1 zmiennej i powtarzamy w pętli aż do momentu, gdy wartość zmiennej i będzie większa niż wartość zmiennej n – wówczas otrzymamy prawidłową wartość silni.

Warunek wykonywania opisanej pętli (zdefiniowany również w kroku 6) jest więc następujący:

$$i \leq n$$



9

Wypisanie wartości zmiennej wynik

Na koniec wypisujemy wartość zmiennej wynik i kończymy wykonywanie algorytmu.

10

Przygotowany program dla zadanej wartości testowej n równej 4 zwróci liczbę 24 (wartość zmiennej wynik).

Gotowy schemat Blockly:

Polecenie 3

Sprawdź swoją wiedzę o algorytmach iteracyjnych, biorąc udział w grze.



Test

Sprawdź swoją wiedzę o algorytmach iteracyjnych, biorąc udział w grze

Poziom trudności:

łatwy

Limit czasu:

7 min

Twój ostatni wynik:

—

Uruchom

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Do przygotowania której z wymienionych funkcjonalności w programie zbędne jest zastosowanie algorytmu iteracyjnego?

☐

Sumowanie n losowych liczb dodatnich, gdzie wartość n podawana jest przez użytkownika.

☐

Obliczanie pola kwadratu o boku długości x podanej przez użytkownika.

☐

Wypisywanie wszystkich liczb pierwszych z zadanego przedziału.

Ćwiczenie 2



Uzupełnij puste miejsca właściwymi liczbami lub wyrażeniami matematycznymi tak, aby przedstawione równości były prawdziwe. Trzy warianty zostały podane dodatkowo i nie pasują do żadnej z luk.

$$\begin{array}{l} 0! = \boxed{} \\ 3! = \boxed{} \\ 7! = \boxed{} \\ 4! = \boxed{} \end{array}$$

$2! \cdot 2!$

$1!$

$6 \cdot 4$

0

6

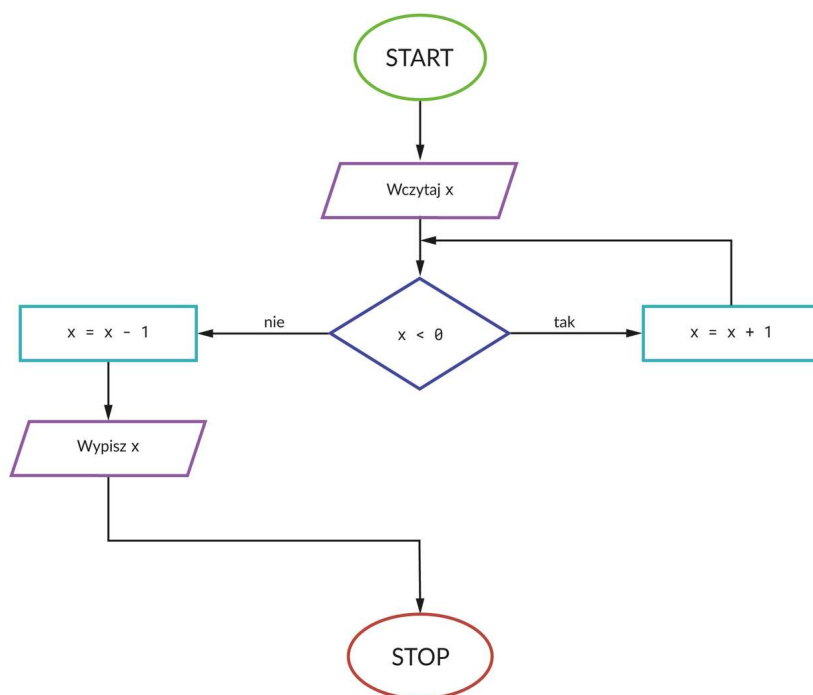
$5! \cdot 6! \cdot 7!$

$5! \cdot 6 \cdot 7$

Ćwiczenie 3



Zapoznaj się ze schematem blokowym i wykonaj polecenie.



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Uzupełnij tekst.

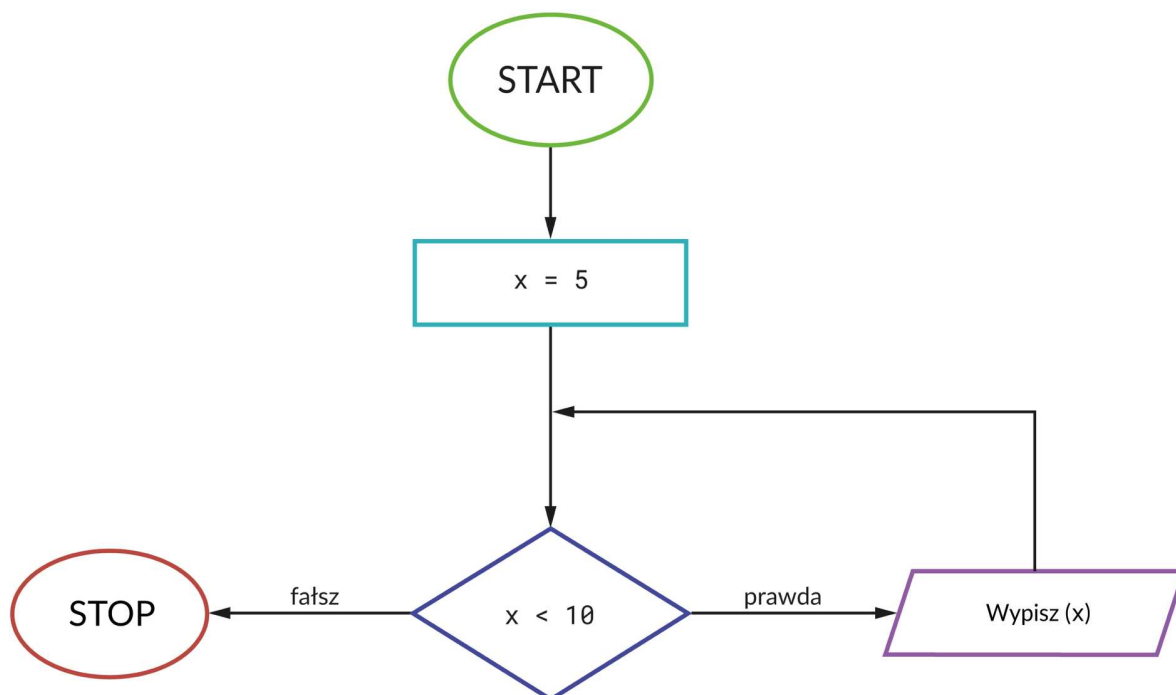
Przy założeniu, że wczytana wartość x wyniesie -5 , liczba wykonań instrukcji $x = x + 1$ wyniesie: .

Przez algorytm zostanie wypisana wartość x . Wyniesie ona: .

Ćwiczenie 4



Wskaż, ile razy zostanie wykonana instrukcja wypisania wartości zmiennej x , wchodząca w skład przedstawionego algorytmu.



☐ 1 raz.

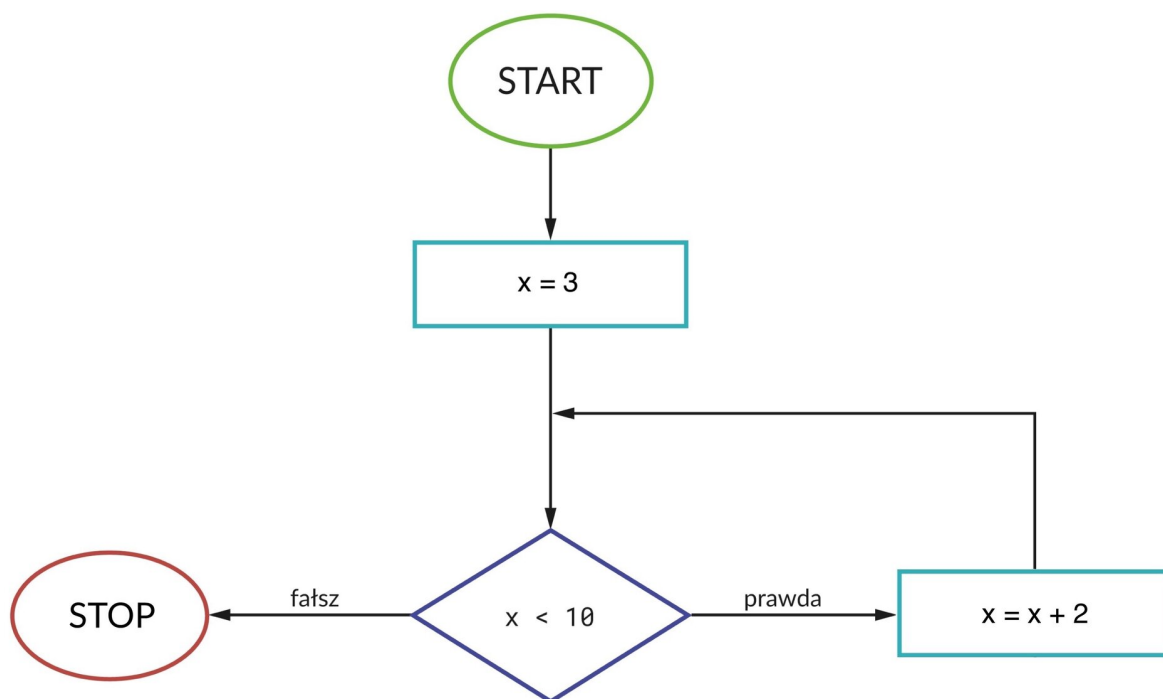
☐ 2 razy.

☐ Nieskończenie wiele razy.

Ćwiczenie 5



Wskaż, ile razy zostanie sprawdzony warunek $x < 10$.



☐ 10

☐ 4

☐ 7

☐ 5

☐ 6

Ćwiczenie 6



Wykonaj poniższe działania i wpisz w puste miejsca uzyskane wyniki. Zakładamy, że a to liczba naturalna większa od 1.

Podczas wprowadzania rozwiązania nie używaj spacji. Otrzymane wyrażenie matematyczne powinno mieć jak najprostszą postać!

• $\frac{8!}{7!} = \frac{8 \cdot 7!}{7!} =$

• $\frac{(a+2)!}{(a+1)!} = \frac{(a+2) \cdot (a+1)!}{(a+1)!} =$

• $\frac{a!}{(a-1)!} =$

• $\frac{a! + (a-1)!}{(a-1)!} =$

Ćwiczenie 7



Przeanalizuj przedstawiony poniżej pseudokod, a następnie uzupełnij zdanie, wstawiając w puste miejsce właściwą wartość.

Pseudokod programu:

```
liczba ← -2
dopóki liczba ≤ 12 wykonuj
    wypisz liczba
    liczba ← liczba + 1
```

Instrukcja wypisania zmiennej `liczba` zostanie wykonana .

14 razy

12 razy

15 razy

13 razy

Ćwiczenie 8



Wskaż, który z poniższych algorytmów przedstawionych za pomocą pseudokodu realizuje algorytm obliczający sumę silni x kolejnych liczb naturalnych według wzoru:

$$\text{suma} = 1! + 2! + \dots + x!$$

Wartość zmiennej x pobierana jest od użytkownika.

Specyfikacja:

Dane:

- x – liczba naturalna większa niż lub równa 2

Wynik:

Program wypisuje obliczoną wartość suma .

```
1 #Początek programu:
2 x ← wprowadź liczbę naturalną
3 suma ← 0
4 silnia ← 1
```



```
1 x ← wprowadź liczbę naturalną
2 suma ← 0
3 silnia ← 1
4
5 dla i = 1,2,...,x wykonuj
6     silnia ← suma * i
7
8 dla j = 1,2,...,x wykonuj
9     suma ← suma + silnia * j
10
11 wypisz suma
```



```
1 x ← wprowadź liczbę naturalną
2 suma ← 0
3 silnia ← 1
4
5 dla i = 1,2,...,x wykonuj
6     silnia ← silnia * i
7     suma ← suma + silnia
8
9 wypisz suma
```



```
1 x ← wprowadź liczbę naturalną
2 suma ← 0
3 silnia ← 1
4
5 dla i = 1,2,...,x wykonuj
6     silnia ← suma * i
7     suma ← suma + silnia
8
9 wypisz suma
```

Dla nauczyciela

Autor: Zespół autorski [Contentplus.pl](https://contentplus.pl) sp. z o.o.

Przedmiot: Informatyka

Temat: Algorytmy iteracyjne – obliczanie silni

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

e) obliczania wartości elementów ciągu metodą iteracyjną i rekurencyjną, w tym wartości elementów ciągu Fibonacciego.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz problem i przedstawisz jego rozwiązanie za pomocą schematu blokowego.
- Wyjaśnisz działanie schematów blokowych na konkretnych przykładach.
- Skonstruujesz własny algorytm iteracyjny, który będzie obliczał silnię.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytmy iteracyjne – obliczanie silni”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Przedstawienie tematu zajęć oraz wspólne z uczniami ustalenie kryteriów sukcesu.

2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu, np.
- na czym polega iteracja?
 - czym jest silnia?
 - jakie figury geometryczne stosujemy w schematach blokowych?
- Chętni lub wybrani uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Gra edukacyjna”, czyta treść polecenia 1 i omawia kolejne kroki rozwiązania. Uczniowie opracowują algorytm za pomocą bloków, a następnie porównują swoje rozwiązania z przedstawionymi w prezentacji.
W kolejnym kroku uczniowie w parach biorą udział w grze edukacyjnej.
2. **Ćwiczenie umiejętności.** Nauczyciel przechodzi do sekcji „Sprawdź się”. Uczniowie indywidualnie rozwiązują ćwiczenia nr 1–8 na czas. Osoba, która poprawnie rozwiąże zadania jako pierwsza, wygrywa, a nauczyciel może nagrodzić ją oceną za aktywność.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Napisz algorytm, który będzie prosił o udzielenie odpowiedzi przez użytkownika, dopóki nie poda on poprawnej odpowiedzi.

Wskazówki metodyczne:

- Treści w sekcji „Gra edukacyjna” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.