



## Algorytmy iteracyjne i liczbowe – zadania maturalne

- Wprowadzenie
- Przeczytaj
- Prezentacja multimedialna
- Sprawdź się
- Dla nauczyciela



## Algorytmy iteracyjne i liczbowe – zadania maturalne

Źródło: Pixabay, domena publiczna.

Kilka ostatnich e-materiałów poświęciliśmy omawianiu algorytmów iteracyjnych. Obecnie skoncentrujemy się na rozwiązywaniu zadań sprawdzających umiejętność analizowania oraz tworzenia algorytmów iteracyjnych.

Za przykład posłużą nam zadania przygotowane przez Centralną Komisję Egzaminacyjną. Przedstawimy również zasady oceniania rozwiązań zadań maturalnych.

Więcej informacji o algorytmach iteracyjnych znajdziesz w e-materiałach:

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

### **Twoje cele**

- Przeanalizujesz sposób rozwiązywania przykładowych zadań maturalnych, dotyczących algorytmów iteracyjnych i liczbowych.
- Rozwiążesz samodzielnie kilka zadań sprawdzających umiejętność zastosowania algorytmów iteracyjnych i liczbowych w rozwiązywaniu konkretnych problemów.
- Prześledzisz zasady oceniania omawianych zadań maturalnych.



# Przeczytaj

---

## Przykładowe zadania maturalne

### Zadanie 1. Liczby Fibonacciego

Liczby Fibonacciego są definiowane w następujący sposób:

$$F_1 = 1 \quad F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad \text{dla } n = 3, 4, \dots$$

Rekurencyjny [algorytm](#), który służy do obliczania wartości  $F_n$  dla dowolnego  $n \geq 1$ , można zapisać następująco:

```
1 funkcja F(n)
2     jeśli n = 1 lub n = 2
3         wynikiem jest 1
4     w przeciwnym razie
5         wynikiem jest F(n - 1) + F(n - 2)
```

Zapisz w wybranej przez siebie notacji (w języku programowania lub w pseudokodzie) algorytm [iteracyjny](#), który służy do obliczania wartości liczby  $F_n$  dla dowolnego  $n \geq 1$ . Algorytm nie może używać tablic.

Zadanie zostało opracowane przez CKE i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2018 roku (poziom rozszerzony).

#### Praca domowa

Przedstaw rozwiązanie zadania w postaci programu w języku C++, Java lub Python.

### Przykładowe rozwiązanie:

#### Specyfikacja problemu:

*Dane:*

- $n$  – liczba naturalna określająca wyraz ciągu Fibonacciego o indeksie  $n$

*Wynik:*

- $fn$  – wartość wyrazu ciągu Fibonacciego o indeksie  $n$

Rozwiązanie przedstawimy w postaci pseudokodu. Na początek zauważmy, że dla każdego wyrazu ciągu Fibonacciego potrzebujemy znać dwa wyrazy go poprzedzające. Dwa początkowe wyrazy są znane, zatem bez problemu możemy wyliczyć trzeci. Podobnie dla wyrazu czwartego skorzystamy z wyrazu drugiego i trzeciego.

Zauważmy, że indeksy wyrazów potrzebnych do zbudowania kolejnego przesunęły się o jeden do przodu. Wystarczy zatem skorzystać z pomocniczej zmiennej, która przechowuje poprzedni z obu wyrazów. Wyrazem poprzednim stanie się obecny wyraz następny, a wyrazem następnym stanie się suma obecnego następnego i zmiennej pomocniczej. Powtarzając procedurę  $n - 2$  razy (dwa pierwsze wyrazy są już znane), możemy obliczyć wartość wyrazu ciągu Fibonacciego o indeksie  $n$ .

```
1 n ← wprowadź liczbę naturalną
2 f1 ← 1
3 f2 ← 1
4 dla i = 3, 4, ..., n wykonuj:
5     pom ← f1
6     f1 ← f2
7     f2 ← f2 + pom
8 zwróć f2
```

### Schemat przyznawania punktów:

2 pkt – za poprawny algorytm (w tym za prawidłowe zapisanie warunków początkowych i iteracji 1 pkt).

1 pkt – za zapamiętanie tylko dwóch poprzednich wyrazów ciągu.

0 pkt – za podanie odpowiedzi błędnej albo za brak odpowiedzi.

### Zadanie 2. Analiza algorytmu

Niech  $n$  będzie nieujemną liczbą całkowitą, a  $T[1..n]$  – tablicą zawierającą  $n$  liczb całkowitych. Dla  $n = 0$  tablica  $T$  jest pusta (nie zawiera żadnego elementu). Przeanalizuj przedstawioną funkcję  $d(x)$ , która rozszerza tablicę  $T$  o liczbę całkowitą  $x$ , a następnie przeprowadza pewną reorganizację zawartości tej tablicy.

```
1 d(x):
2     n ← n + 1
3     T[n] ← x
4     s ← n
5     dopóki ((s div 2) ≥ 1) oraz (T[s] > T[s div 2]) wykonuj
```

```

6     pom ← T[s]
7     T[s] ← T[s div 2]
8     T[s div 2] ← pom
9     s ← s div 2

```

**Uwaga:** w tym zadaniu przyjmujemy, że:

- tablica T może być powiększana;
- jeśli wartość lewego argumentu operatora oraz jest to fałsz, to wartość prawego argumentu nie jest wyliczana;
- div jest operatorem oznaczającym część całkowitą z dzielenia.

Uzupełnij tabelę – wpisz zawartość tablicy T po wykonaniu  $d(x)$  z podanym parametrem x:

n	T[1..n]	x	T po wykonaniu d(x)
4	26, 3, 5, -4	5	26, 5, 5, -4, 3
4	36, 15, 17, 3	-5	?
7	27, 6, 13, 4, -3, -2, -3	30	?

Zadanie zostało opracowane przez CKE i pojawiło się na egzaminie maturalnym z informatyki w 2021 roku (poziom rozszerzony). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

#### Praca domowa

Przedstaw rozwiązanie zadania w postaci programu w języku C++, Java lub Python.

#### Przykładowe rozwiązanie:

#### Specyfikacja problemu:

*Dane:*

- n – liczba naturalna
- T[1..n] – tablica zawierająca n liczb całkowitych

*Wynik:*

- T[1..n] – tablica liczb całkowitych powstała po wykonaniu  $d(x)$

Zgodnie z przedstawionym algorytmem wstawiamy element x na koniec tablicy T, jej nowy rozmiar zapisujemy do pomocniczej zmiennej s, a następnie, dopóki  $s \text{ div } 2$  jest większe lub równe 1 oraz element o indeksie s jest większy od elementu o indeksie  $s \text{ div } 2$ , to zamieniamy oba elementy miejscami i dzielimy s całkowitoliczbowo przez 2.

Dla drugiego wiersza  $s$  wynosi 5, zatem pierwsza iteracja sprawdzi elementy o indeksach 5 oraz 2. Element  $T[5]$  jest jednak mniejszy od  $T[2]$ , czyli warunek nie jest spełniony. Na koniec tablicy  $T$  zostanie jedynie wstawiony argument  $x$  o wartości  $-5$ .

36, 15, 17, 3, -5
-------------------

Dla trzeciego wiersza  $s$  wynosi 8, zatem najpierw sprawdzone zostaną elementy o indeksach 8 oraz 4. Element  $T[4]$  jest mniejszy, więc zostaną zamienione:

27, 6, 13, 4, -3, -2, -3, 30	27, 6, 13, 30, -3, -2, -3, 4
------------------------------	------------------------------

W drugiej iteracji sprawdzamy elementy o indeksach 4 oraz 2. Ponownie zachodzi zamiana:

27, 6, 13, 30, -3, -2, -3, 4	27, 30, 13, 6, -3, -2, -3, 4
------------------------------	------------------------------

W ostatniej iteracji porównujemy dwa pierwsze elementy tablicy, dla których również spełniony jest warunek pętli:

27, 30, 13, 6, -3, -2, -3, 4	30, 27, 13, 6, -3, -2, -3, 4
------------------------------	------------------------------

Końcowa tabela przedstawia się następująco:

n	T[1..n]	x	T po wykonaniu d(x)
4	26, 3, 5, -4	5	26, 5, 5, -4, 3
4	36, 15, 17, 3	-5	36, 15, 17, 3, -5
7	27, 6, 13, 4, -3, -2, -3	30	30, 27, 13, 6, -3, -2, -3, 4

### Schemat przyznawania punktów:

2 pkt – za poprawną odpowiedź w obu wierszach.

1 pkt – za poprawną odpowiedź w jednym wierszu.

0 pkt – za podanie odpowiedzi niepoprawnej albo za brak odpowiedzi.

## Słownik

### algorytm

przepis postępowania prowadzący do rozwiązania ustalonego problemu, określający ciąg czynności elementarnych, które należy w tym celu wykonać

### iteracja

technika programowania polegająca na powtarzaniu tych samych operacji w pętli określonej liczbą razy lub do momentu, aż zostanie spełniony zadany warunek

### schemat oceniania

dokument, który precyzyjnie określa, które elementy rozwiązania zadań są sprawdzane i w jaki sposób przyznawane są punkty

# Prezentacja multimedialna

---

## Zadanie 3: Księgarnia

Pan Kowalski prowadzi małą rodzinną księgarnię. Codziennie rano odbiera od kuriera dostawę nowych książek, z których część wystawia od razu na sprzedaż. Po pewnym czasie zauważył, że liczba książek dostarczonych oraz liczba książek sprzedanych danego dnia zależą od dnia miesiąca i są opisane następującym pseudokodem:

```
1 liczba_ksiazek_dostarczonych ← (dzien_miesiaca * 3) mod 10
2
3 jeżeli dzien_miesiaca mod 5 = 0 wykonaj:
4     liczba_ksiazek_sprzedanych ← liczba_ksiazek_dostarczonych div
5
6 w przeciwnym wypadku wykonaj:
7     liczba_ksiazek_sprzedanych ← liczba_ksiazek_dostarczonych div
```

### Uwaga!

Operator `div` oznacza dzielenie całkowitoliczbowe, natomiast `mod` resztę z dzielenia.

### Polecenie 1

Na podstawie przedstawionych zależności, napisz w pseudokodzie lub wybranym języku programowania algorytm, który wypisze, po ilu dniach (począwszy od początku miesiąca) pan Kowalski sprzeda co najmniej 100 książek oraz wypisze, ile w tym czasie będzie dni, w których sprzeda więcej książek niż przyjmie w dostawie.

### Uwaga!

W zapisie algorytmu możesz wykorzystać tylko operacje arytmetyczne: dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, resztę z dzielenia oraz porównywanie liczb, instrukcje sterujące i przypisania do zmiennych lub samodzielnie napisane funkcje zawierające wymienione operacje.

## Rozwiązanie

Rozwiązanie przedstawimy w postaci pseudokodu, ponieważ na egzaminie maturalnym można korzystać z wybranego języka programowania: C++, Java lub Python.



Algorytm może pomóc podczas pracy z księgozbiorem, np. w planowaniu kalendarza dostaw i zakupów w księgarni.  
Źródło: pixabay.com, domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

Na początku wczytujemy dane z zadania. Numerowanie dni miesiąca zaczynamy od zera, ponieważ w przeciwnym wypadku po zakończeniu pętli otrzymalibyśmy pierwszy dzień po przekroczeniu 100 sprzedanych książek (każdy przebieg pętli jest wtedy równoznaczny z danym dniem):

```
1 dzien_miesiaca ← 0
2 suma_sprzedanych ← 0
3 dni_lepszej_sprzedazy ← 0
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

Na samym początku pętli zwiększamy liczbę dni miesiąca. Następnie obliczamy liczby dostarczonych i sprzedanych książek w danym dniu, zgodnie z zależnością opisaną w poleceniu za pomocą pseudokodu.

Pętla ma się powtarzać do momentu przekroczenia 100 sprzedanych książek:

```
1 dzien_miesiaca ← 0
2 suma_sprzedanych ← 0
3 dni_lepszej_sprzedazy ← 0
4
5 dopóki suma_sprzedanych <
  100 wykonuj:
6
7     dzien_miesiaca ←
  dzien_miesiaca + 1
8
9     liczba_dostarczonych ←
  (dzien_miesiaca * 3) mod
  10
10
11     jeżeli dzien_miesiaca
  mod 5 = 0 wykonaj:
12         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + (dzien_miesiaca + 2)
  mod 11
13     w przeciwnym wypadku
  wykonaj:
14         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + dzien_miesiaca
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

3

Po wyliczeniu wielkości dostawy oraz sprzedaży sprawdzamy, czy książek sprzedanych jest więcej niż dostarczonych.

Jeśli tak, zwiększamy o jeden zmienną `dni_lepszej_sprzedazy`.

```
1 dzien_miesiaca ← 0
```

```

2 suma_sprzedanych ← 0
3 dni_lepszej_sprzedazy ← 0
4
5 dopóki suma_sprzedanych <
  100 wykonuj:
6
7     dzien_miesiaca ←
  dzien_miesiaca + 1
8
9     liczba_dostarczonych ←
  (dzien_miesiaca * 3) mod
  10
10
11     jeżeli dzien_miesiaca
  mod 5 = 0 wykonaj:
12         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + (dzien_miesiaca + 2)
  mod 11
13     w przeciwnym wypadku
  wykonaj:
14         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + dzien_miesiaca
15
16     jeżeli
  liczba_dostarczonych <
  liczba_sprzedanych
  wykonaj:
17
  dni_lepszej_sprzedazy ←
  dni_lepszej_sprzedazy + 1

```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

Na końcu zwiększamy sumę sprzedanych książek o liczbę sprzedanych sztuk w danym dniu.

Instrukcje pętli kończą się i ponownie sprawdzany jest początkowy warunek:

```
1  dzien_miesiaca ← 0
2  suma_sprzedanych ← 0
3  dni_lepszej_sprzedazy ← 0
4
5  dopóki suma_sprzedanych <
   100 wykonuj:
6
7      dzien_miesiaca ←
   dzien_miesiaca + 1
8
9      liczba_dostarczonych ←
   (dzien_miesiaca * 3) mod
   10
10
11     jeżeli dzien_miesiaca
   mod 5 = 0 wykonaj:
12         liczba_sprzedanych
   ← liczba_dostarczonych div
   2 + (dzien_miesiaca + 2)
   mod 11
13     w przeciwnym wypadku
   wykonaj:
14         liczba_sprzedanych
   ← liczba_dostarczonych div
   2 + dzien_miesiaca
15
16     jeżeli
   liczba_dostarczonych <
   liczba_sprzedanych
   wykonaj:
17
18     dni_lepszej_sprzedazy ←
   dni_lepszej_sprzedazy + 1
19
20     suma_sprzedanych ←
   suma_sprzedanych +
   liczba_sprzedanych
```



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

5

Po zakończeniu działania pętli wypisujemy dzień miesiąca, w którym przekroczyliśmy liczbę 100 sprzedanych książek oraz liczbę dni, w których sprzedaż była większa od dostawy.

```
1 dzien_miesiaca ← 0
2 suma_sprzedanych ← 0
3 dni_lepszej_sprzedazy ← 0
4
5 dopóki suma_sprzedanych <
  100 wykonuj:
6
7     dzien_miesiaca ←
  dzien_miesiaca + 1
8
9     liczba_dostarczonych ←
  (dzien_miesiaca * 3) mod
  10
10
11     jeżeli dzien_miesiaca
  mod 5 = 0 wykonaj:
12         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + (dzien_miesiaca + 2)
  mod 11
13     w przeciwnym wypadku
  wykonaj:
14         liczba_sprzedanych
  ← liczba_dostarczonych div
  2 + dzien_miesiaca
15
16     jeżeli
  liczba_dostarczonych <
  liczba_sprzedanych
  wykonaj:
17
18         dni_lepszej_sprzedazy ←
  dni_lepszej_sprzedazy + 1
19
20     suma_sprzedanych ←
  suma_sprzedanych +
```

liczba\_sprzedanych  
20  
21 wypisz dzien\_miesiaca,  
dni\_lepszej\_sprzedazy

6



Salon Klubu Międzynarodowej Prasy i Książki w Domach Towarowych „Centrum” na Ścianie Wschodniej w Warszawie.

Źródło: Wikimedia Commons, commons.wikimedia.org, dostęp 08.05.2024, licencja: domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P2B9WGHG6>

W tabeli przedstawione są liczby dostarczonych i sprzedanych książek (dane uzyskane w wyniku działania algorytmu), a także suma sprzedaży i liczba dni, w których wysokość sprzedaży była wyższa od dostawy w zależności od dnia miesiąca.

Dzień miesiąca	Liczba dostarczonych książek	Liczba sprzedanych książek	Suma sprzedanych książek	Lic l sp d
1	3	2	2	
2	6	5	7	

Dzień miesiąca	Liczba dostarczonych książek	Liczba sprzedanych książek	Suma sprzedanych książek	Liczba dni sprzedaży
3	9	7	14	
4	2	5	19	
5	5	9	28	
6	8	10	38	
7	1	7	45	
8	4	10	55	
9	7	12	67	
10	0	1	68	
11	3	12	80	
12	6	15	95	
13	9	17	112	

Księgarnia Kowalskiego osiągnęła sprzedaż 100 książek 13. dnia miesiąca. W tym okresie było 10 dni, w których sprzedaż była większa od dostawy.

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

## Polecenie 2

Dodaj do swojego programu komentarze tak, żeby był zrozumiały dla osoby, która nie potrafi programować.

# Sprawdź się

---

Pokaż ćwiczenia:   

## Zadanie 4: Test

W przedstawionym algorytmie  $n$  jest nieujemną liczbą całkowitą,  $\text{mod}$  to operator reszty z dzielenia,  $\text{div}$  to operator dzielenia całkowitego.

```
1  $w \leftarrow 0$ 
2 dopóki  $n \neq 0$  wykonuj
3      $w \leftarrow w + (n \text{ mod } 10)$ 
4      $n \leftarrow n \text{ div } 10$ 
```

### Ćwiczenie 1



Zaznacz wszystkie prawdziwe stwierdzenia.

Po wykonaniu algorytmu dla liczby  $n$  wartością zmiennej  $w$  jest suma cyfr liczby  $n$  w zapisie dziesiętnym.

Po wykonaniu algorytmu dla  $n = 11111$  zmienna  $w$  przyjmuje wartość 5.

Po wykonaniu algorytmu dla  $n = 45778$  zmienna  $w$  przyjmuje wartość 30.

Podczas wykonywania algorytmu dla  $n = 1234$  w kolejnych iteracjach pętli dopóki, zmienna  $w$  przyjmuje wartości 1, 3, 6, 10.

### Schemat przyznawania punktów:

1 pkt – za poprawną odpowiedź.

0 pkt – za odpowiedź niepełną lub niepoprawną albo za brak odpowiedzi.

Zadanie zostało opracowane przez CKE i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2020 roku (poziom rozszerzony, część 1). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

## Ćwiczenie 2



Popraw błędne odpowiedzi z poprzedniego zadania.

### Praca domowa

Zapisz algorytm zapisany za pomocą pseudokodu w Ćwiczeniu 4, wykorzystując wybrany język programowania.

1

1

## Zadanie 5: Analiza algorytmu

Rozważamy następujący algorytm:

### Specyfikacja problemu:

*Dane:*

- $n$  – liczba całkowita dodatnia

*Wynik:*

- $p$  – liczba całkowita dodatnia

```
1 p ← 1
2 q ← n
3 dopóki p < q wykonuj
4     s ← (p + q) div 2
5     (*) jeżeli s*s*s < n wykonaj
6         p ← s + 1
7     w przeciwnym wypadku
8         q ← s
```

**Uwaga:** zapis  $\text{div}$  oznacza dzielenie całkowite.

### Ćwiczenie 3



Podaj wynik działania algorytmu dla wskazanych w tabeli wartości  $n$ :

$n$	$p$
28	<input type="text"/>
64	<input type="text"/>
80	<input type="text"/>

Schemat przyznawania punktów:

3 pkt – za prawidłową odpowiedź w trzech wierszach.

2 pkt – za prawidłową odpowiedź w dwóch wierszach.

1 pkt – za prawidłową odpowiedź w jednym wierszu.

0 pkt – za podanie odpowiedzi błędnej albo brak odpowiedzi.

#### Ćwiczenie 4



Podaj najmniejszą oraz największą liczbę  $n$ , dla której wynikiem działania algorytmu będzie  $p = 10$ .

Największa liczba  $n$ :

Najmniejsza liczba  $n$ :

#### Schemat przyznawania punktów:

2 pkt – za dwie poprawne odpowiedzi.

1 pkt – za jedną poprawną odpowiedź.

0 pkt – za podanie odpowiedzi błędnej albo brak odpowiedzi.

#### Ćwiczenie 5



Dokończ zdanie. Wybierz i zaznacz właściwą odpowiedź spośród podanych.

Dla każdej liczby całkowitej  $n > 1$  instrukcja oznaczona w algorytmie symbolem (\*) wykona się...

więcej niż  $n + 1$ , ale mniej niż  $2n$  razy.

mniej niż  $2 * \log_2(n)$  razy.

więcej niż  $n^2$  razy.

więcej niż  $n/2$ , ale mniej niż  $n$  razy.

## Ćwiczenie 6



Dokończ zdanie. Wybierz i zaznacz właściwą odpowiedź spośród podanych.

Dla każdej liczby całkowitej  $n > 1$  instrukcja oznaczona w algorytmie symbolem (\*) wykonana się...

więcej niż  $n + 1$ , ale mniej niż  $2n$  razy.

mniej niż  $2 * \log_2(n)$  razy.

więcej niż  $n^2$  razy.

więcej niż  $n/2$ , ale mniej niż  $n$  razy.

### Schemat przyznawania punktów:

1 pkt – za poprawną odpowiedź.

0 pkt – za podanie odpowiedzi błędnej albo brak odpowiedzi.

Zadanie zostało opracowane przez CKE i pojawiło się na egzaminie maturalnym z informatyki w maju 2018 roku (poziom rozszerzony, część 1). Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

### Zadanie 6: Sitko

Rozważmy następującą procedurę, której parametrem jest dodatnia liczba całkowita  $n$ .

```
1 Procedura Sitko(n)
2   Czyjest[1..n]
3   dla i = 1, 2, ..., n wykonuj
4     Czyjest[i] ← fałsz
5     j ← 1
6     dopóki n > j * j wykonuj
7       j ← j + 1
8   dla i = 2, 3, ..., j wykonuj
9     kw ← i * i
10    poz ← kw
11    dopóki poz ≤ n wykonuj
12    (*) Czyjest[poz] ← prawda
```

### Ćwiczenie 7



Uzupełnij poniższą tabelę – wpisz wartości zmiennych  $j$  oraz  $\text{Czyjest}[k]$  po wykonaniu  $\text{Sitko}(n)$ .

n	k	j	Czyjest[k]
10	9	4	prawda
10	5	<input type="text"/>	<input type="text"/>
100	10	<input type="text"/>	<input type="text"/>
100	75	<input type="text"/>	<input type="text"/>

Uwaga, w kolumnie  $\text{Czyjest}[k]$  należy wpisać prawda lub fałsz. Inne zapisy odpowiedzi zostaną uznane za błędne.

### Schemat przyznawania punktów:

1 pkt – za poprawnie wypełnioną tabelę.

0 pkt – za podanie odpowiedzi błędnej albo brak odpowiedzi.

### Ćwiczenie 8



Rozważmy działanie  $\text{Sitko}(100)$ . Podaj liczbę wykonań instrukcji w wierszu oznaczonym (\*) – dla wartości zmiennej  $i$  wskazanych w tabeli.

i	liczba wykonań wiersza (*)
2	25
3	<input type="text"/>
5	<input type="text"/>
9	<input type="text"/>

## Ćwiczenie 9



Spośród poniższych wartości zaznacz te, które są większe niż łączna liczba wykonań instrukcji z wiersza (\*) w trakcie wykonywania procedury Sitko(100).

100

$\sqrt{100}$

$\ln(100)$

$100 \times \left( \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{10^2} \right)$

### Schemat przyznawania punktów:

1 pkt – za poprawnie wskazane wartości.

0 pkt – za podanie odpowiedzi błędnej albo brak odpowiedzi.

Zadanie zostało opracowane przez CKE i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2017 roku (poziom rozszerzony, część 1).

# Dla nauczyciela

---

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

**Przedmiot:** Informatyka

**Temat: Algorytmy iteracyjne i liczbowe – zadania maturalne**

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

**Grupa docelowa:**

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

**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:

1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).

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

a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW,

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

**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 sposób rozwiązywania przykładowych zadań maturalnych, dotyczących algorytmów iteracyjnych i liczbowych.
- Rozwiążesz samodzielnie kilka zadań sprawdzających umiejętność zastosowania algorytmów iteracyjnych i liczbowych w rozwiązywaniu konkretnych problemów.
- Prześledzisz zasady oceniania omawianych zadań maturalnych.

### **Strategie nauczania:**

- konstruktywizm;
- konektywizm.

### **Metody i techniki nauczania:**

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

### **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;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

### **Przebieg lekcji**

#### **Przed lekcją:**

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytmy iteracyjne i liczbowe – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Prezentacja multimedialna”.

#### **Faza wstępna:**

1. Nauczyciel wprowadza uczniów szczegółowo w temat lekcji i jej cele. Może posłużyć się wyświetloną na tablicy zawartością sekcji „Wprowadzenie”.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu i programowania, np. :
  - do jakich działań możemy wykorzystać algorytmy iteracyjne i liczbowe?
  - czy wiecie jak wygląda schemat wystawiania ocen na egzaminie maturalnym?Chętni uczniowie udzielają na nie odpowiedzi.

#### **Faza realizacyjna:**

1. **Praca z tekstem.** Nauczyciel wyświetla zawartość sekcji „Przeczytaj”. Na forum klasy uczniowie analizują przedstawione w niej rozwiązania zadań 1 i 2.
2. **Praca z multimedium.** Uczniowie w grupach analizują zadanie przedstawione w sekcji „Prezentacja multimedialna”. Następnie opracowują rozwiązanie, korzystając z wybranego języka programowania. Chętne lub wybrane osoby przedstawiają swój kod na forum klasy.
3. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że w kolejnym kroku będą rozwiązywać ćwiczenia nr 1-5 z sekcji „Sprawdź się”. Każdy z uczniów robi to samodzielnie. Po ustalonym czasie wybrani uczniowie przedstawiają rozwiązania. Nauczyciel w razie potrzeby koryguje odpowiedzi, dopowiada istotne informacje, udziela uczniom informacji zwrotnej.

#### **Faza podsumowująca:**

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.

#### **Praca domowa:**

1. Uczniowie wykonują ćwiczenia 6-8 z sekcji „Sprawdź się”.

#### **Wskazówki metodyczne:**

- Multimedia w sekcjach: „Przeczytaj”, „Prezentacja multimedialna”, „Sprawdź się” można wykorzystać jako materiał służący powtórzeniu materiału.