

# Test wiedzy algorytmicznej

XIX OIJ, zawody I stopnia, tura testowa  
24 października 2024



Olimpiada  
Informatyczna  
Juniorów

Test składa się z 20 pytań. Odpowiedzi należy zaznaczać na karcie odpowiedzi. Pytania są dwóch typów:

- pytania wielokrotnego wyboru – należy w nich zaznaczyć znakami  $\times$  kratki obok wszystkich prawidłowych odpowiedzi, a pozostałe kratki obok nieprawidłowych odpowiedzi pozostawić puste,
- pytania krótkiej odpowiedzi – należy w nich wpisać odpowiedź, zawsze jest to liczba naturalna lub krótki napis. Jeżeli odpowiedzią jest liczba naturalna to należy ją zapisać z użyciem cyfr arabskich, w najprostszej postaci w systemie dziesiętkowym (np. jeśli poprawna odpowiedź to 4, to odpowiedzi 4.0,  $\frac{8}{2}$ ,  $\sqrt{16}$ , IV,  $100_2$ ,  $2^2$  nie będą akceptowane). Jeżeli odpowiedzią jest krótki napis, należy go zapisać dokładnie, bez zmiany wielkości znaków, bez dodatkowych odstępów, bez dodatkowej interpunkcji kończącej zdanie (np. jeśli poprawna odpowiedź to kajak to odpowiedzi k a j a k, k-a-j-a-k, Kajak, Odpowiedź to: kajak., KAJAK nie będą akceptowane).

W razie pomyłki przy wypełnianiu karty odpowiedzi, poproś pilnującego test o nową, pustą kartę odpowiedzi.

Oceniana jest tylko poprawność odpowiedzi. Nie należy podawać żadnych uzasadnień.

Na rozwiązanie testu masz 100 minut. Na końcu testu znajdują się kartki brudnopisu. Używanie urządzeń elektronicznych, w tym kalkulatorów (nawet prostych) jest zakazane.

Za każdą w pełni poprawną odpowiedź na pytanie otrzymasz 10 punktów. Za błędną odpowiedź lub brak odpowiedzi nie przyznaje się punktów. Możesz założyć, że w pytaniach na teście podane kody w C++ i w Pythonie są równoważne.

Prawidłowe odpowiedzi wraz z wyjaśnieniami opublikowane zostaną na stronie <https://oij.edu.pl> we wtorek 29 października 2024 o godzinie 13.00. Wtedy też, po zalogowaniu do systemu SIO2, będzie można przeglądać wyniki testu.

Pamiętaj, że oprócz testu, elementem I etapu zawodów OIJ są zadania programistyczne. Możesz je znaleźć w systemie SIO2. Zadania programistyczne można rozwiązywać do 8 stycznia 2025. O kwalifikacji do II etapu zawodów decyduje suma punktów uzyskanych na teście oraz z zadań programistycznych. Każdy uczeń, który weźmie udział w teście lub prześle rozwiązanie przynajmniej jednego zadania programistycznego staje się uczestnikiem Olimpiady i w zależności od uzyskanego sumarycznego wyniku może zostać zakwalifikowany do II etapu. Nie jest konieczne rozwiązanie wszystkich zadań ani odpowiedzenie na wszystkie pytania.



1. Rozważmy poniższy fragment programu:

wersja C++

```
for (int i = 0; i < 10; i += 2)
    cout << "*";
```

wersja Python

```
for i in range(0, 10, 2):
    print('*', end='')
```

Ile gwiazdek zostanie wypisanych przez powyższy fragment programu?

2. Rozważmy poniższą funkcję:

wersja C++

```
int maksimum(int x, int y, int z) {
    if ((x >= y) && (x >= z))
        return x;
    if (...)
        return y;
    return z;
}
```

wersja Python

```
def maksimum(x, y, z):
    if (x >= y) and (x >= z):
        return x
    if ...:
        return y
    return z
```

Jak należy uzupełnić warunek w powyższym kodzie (w miejscu trzech kropek), aby funkcja zwracała największą z trzech podanych do niej liczb?

- A.  $y \geq x$
- B.  $y \geq z$
- C.  $x \geq z$
- D.  $y < z$

3. Rozważmy poniższy fragment programu:

wersja C++

```
int masa = 90;
double wzrost = 2.0;
double bmi = masa / (wzrost * wzrost);
if (bmi < 20.0)
    cout << "Niedowaga.\n";
else if (bmi <= 25.0)
    cout << "Waga w normie.\n";
else if (bmi <= 30.0)
    cout << "Nadwaga.\n";
else
    cout << "Otylosc.\n";
```

wersja Python

```
masa = 90
wzrost = 2.0
bmi = masa / (wzrost * wzrost)
if bmi < 20.0:
    print('Niedowaga.')
```

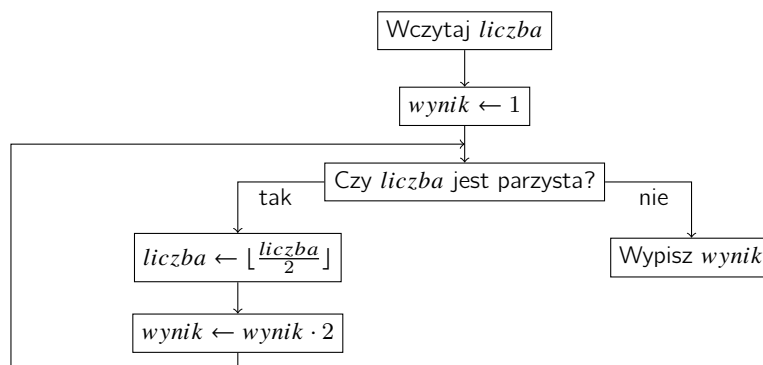
Jaki napis wypisze powyższy fragment programu?

- A. Niedowaga.
- B. Waga w normie.
- C. Nadwaga.
- D. Otylosc.

4. Ciąg Fibonacciego zdefiniowany jest w następujący sposób:  $F_0 = F_1 = 1$  oraz  $F_n = F_{n-1} + F_{n-2}$  dla liczb naturalnych  $n \geq 2$ . Innymi słowy: zerowy i pierwszy wyraz ciągu to 1, zaś każdy kolejny wyraz jest sumą dwóch poprzednich. Kilka pierwszych wyrazów ciągu Fibonacciego to 1, 1, 2, 3, 5, 8. Ile jest dwucyfrowych liczb w ciągu Fibonacciego?



5. Rozważmy poniższy schemat blokowy. Co wypisze program, jeżeli wprowadzimy na wejściu wartość 20? Dla przypomnienia: symbol  $\lfloor \cdot \rfloor$  oznacza zaokrąglenie w dół do najbliższej liczby całkowitej (przykładowo:  $\lfloor \frac{8}{3} \rfloor = 2$ ).



6. Rozważmy poniższą funkcję:

wersja C++

```
int potega(int x, int y) {
    if (y == 0)
        return 1;
    return ...;
}
```

wersja Python

```
def potega(x, y):
    if y == 0:
        return 1
    return ...
```

Jak należy uzupełnić powyższy kod (w miejscu trzech kropek), aby wartość zwrócona przez wywołanie  $\text{potega}(x, y)$  była równa  $x^y$  ( $x$  do potęgi  $y$ ) dla wszystkich liczb naturalnych  $x, y$  spełniających warunek  $1 \leq x, y \leq 9$ ?

- A.  $x * y$
- B.  $x * \text{potega}(x, y - 1)$
- C.  $\text{potega}(x, y)$
- D.  $x \wedge y$

7. Rozważmy poniższą funkcję:

wersja C++

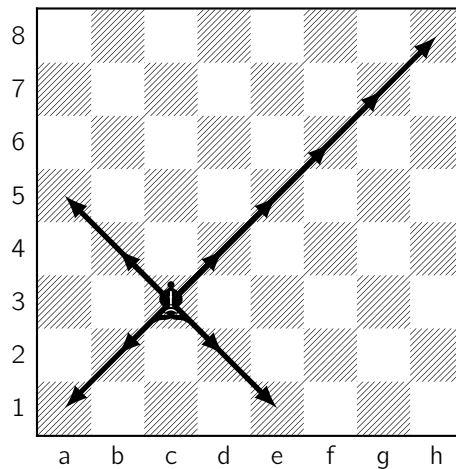
```
int przekształcenie(int n) {
    return (2 * n) / 5;
}
```

wersja Python

```
def przekształcenie(n):
    return (2 * n) // 5
```

Ile jest nieujemnych liczb całkowitych  $n$  nie przekraczających 100, które podane jako argument funkcji  $\text{przekształcenie}$  dają w wyniku wartość 4?

8. Rozważmy standardową szachownicę  $8 \times 8$  oraz figurę gońca, który atakuje wszystkie pola wzdłuż obu przekątnych, w dowolnej odległości, z wyjątkiem pola, na którym sam jest (jak pokazano na rysunku poniżej).



Przykład obrazujący, że gońiec ustawiony na polu c3 atakuje 11 pól. Zakładamy, że gońiec nie atakuje pola c3.

Niech  $A$  oznacza najmniejszą, a  $B$  największą liczbę pól jakie może atakować gońiec ustawiony na pewnym polu szachownicy, na której nie znajdują się żadne inne bierki. Ile wynosi wartość  $A + B$ ?

9. Rozważmy poniższą funkcję:

wersja C++

```
string konstruuuj(string slowo, vector<int> pozycje) {
    string wynik = "";
    for (int i = 0; i < pozycje.size(); i++)
        wynik += slowo[pozycje[i]];
    return wynik;
}
```

Założmy następującą definicję:

(w C++)

```
vector<int> pozycje({ 3, 0, 7, 6 });
```

(w Pythonie)

```
pozycje = [3, 0, 7, 6]
```

Jaki będzie wynik wywołania `konstruuuj("olimpiada", pozycje)`?

wersja Python

```
def konstruuuj(slowo, pozycje):
    wynik = ''
    for i in range(len(pozycje)):
        wynik += slowo[pozycje[i]]
    return wynik
```

10. Rozważmy poniższą funkcję:

wersja C++

```
void zmieniaj(long long x, long long y, int delta) {
    while (x != y) {
        cout << x << "\n";
        x += delta;
        y -= delta;
    }
}
```

wersja Python

```
def zmieniaj(x, y, delta):
    while x != y:
        print(x)
        x += delta
        y -= delta
```

Wybierz wszystkie wywołania, które (na typowym komputerze osobistym) zakończą się w czasie krótszym niż sekunda.

- A. zmieniaj(5, 10, 1)
- B. zmieniaj(8, 20, 2)
- C. zmieniaj(20, 8, 2)
- D. zmieniaj(3, 13, 5)

11. Rozważmy funkcję  $R: \mathbb{N}_+ \rightarrow \mathbb{Z}$  zdefiniowaną w następujący sposób:

$$R(n) = \begin{cases} R(\lfloor \frac{n}{2} \rfloor) + 1 & \text{gdy } n > 1 \\ 0 & \text{gdy } n = 1 \end{cases}$$

Jaka jest najmniejsza wartość  $n$ , dla której  $R(n) = 5$ ? Dla przypomnienia: symbol  $\lfloor \cdot \rfloor$  oznacza zaokrąglenie w dół do najbliższej liczby całkowitej (przykładowo:  $\lfloor \frac{8}{3} \rfloor = 2$ ).

12. Rozważmy poniższe funkcje:

wersja C++

```
int f(int n) {
    return 2 * n + 1;
}

int g(int n) {
    return f(f(n));
}

int h(int n) {
    int a = ...;
    int b = ...;
    return a * n + b;
}
```

wersja Python

```
def f(n):
    return 2 * n + 1

def g(n):
    return f(f(n))

def h(n):
    a = ...
    b = ...
    return a * n + b
```

Jakimi liczbami można uzupełnić ciało funkcji  $h$  w miejscu kropek, żeby dla każdej liczby całkowitej  $n$  (z zakresu  $-100 \leq n \leq 100$ ) wartości  $g(n)$  oraz  $h(n)$  były równe? Jako odpowiedź podaj wartość sumy  $a + b$ .



13. Rozważmy poniższy fragment programu:

wersja C++

```
string napis = "oolimpiadaaa";
int wynik = 0;
for (char znak1 : napis)
    for (char znak2 : napis)
        if (znak1 == znak2)
            wynik++;
cout << wynik << "\n";
```

wersja Python

```
napis = 'oolimpiadaaa'
wynik = 0
for znak1 in napis:
    for znak2 in napis:
        if znak1 == znak2:
            wynik += 1
print(wynik)
```

Jaka liczba zostanie wypisana po wykonaniu powyższego fragmentu programu?

14. Rozważmy poniższą funkcję:

wersja C++

```
int ile_liter(string napis) {
    vector<bool> bylo(26, false);
    for (char znak : napis)
        bylo[znak - 'a'] = true;
    int wynik = 0;
    for (int i = 0; i < 26; i++)
        if (bylo[i])
            wynik++;
    return wynik;
}
```

wersja Python

```
def ile_liter(napis):
    bylo = [False] * 26
    for znak in napis:
        bylo[ord(znak) - ord('a')] = True
    wynik = 0
    for i in range(26):
        if bylo[i]:
            wynik += 1
    return wynik
```

Wybierz wszystkie napisy, które podane jako argument funkcji `ile_liter` dadzą wynik 3.

- A. kajak
- B. ioi
- C. oij
- D. olimpiada

15. Jaką sumę cyfr miałyby liczba  $8251325_{(9)}$  (liczba ta zapisana jest w systemie dziewiętkowym), gdyby zapisać ją w systemie trójkowym?

16. Niech  $s(n)$  jest funkcją, która zwraca sumę cyfr liczby  $n$  w zapisie dziesiętnym. Na przykład  $s(123) = 1 + 2 + 3 = 6$ . Podaj najmniejszą liczbę naturalną, dla której  $s(s(n)) = 10$ .

17. Rozważmy poniższą funkcję:

wersja C++

```
void wypisuj(int n) {
    if (n == 0)
        return;
    cout << "*";
    for (int i = 0; i < n; i++)
        wypisuj(i);
}
```

wersja Python

```
def wypisuj(n):
    if n == 0:
        return
    print('*', end='')
    for i in range(n):
        wypisuj(i)
```

Ile gwiazdek zostanie wypisanych po wywołaniu `wypisuj(8)`?

## 18. Rozważmy poniższe funkcje:

### wersja C++

```
string na_cyfre(int cyfra, string znaki) {
    char jednosc = znaki[0];
    char piatek = znaki[1];
    char dziesiatka = znaki[2];
    if (cyfra == 4) return string(1, jednosc) + piatek;
    if (cyfra == 9) return string(1, jednosc) + dziesiatka;
    string wynik = "";
    if (cyfra >= 5) wynik = piatek;
    for (int i = 0; i < cyfra % 5; i++) wynik += jednosc;
    return wynik;
}

string na_rzymska(int n) {
    string wynik = "";
    for (int i = 0; i < n / 1000; i++) wynik += "M";
    wynik += na_cyfre(n % 1000 / 100, "CDM");
    wynik += na_cyfre(n % 100 / 10, "XLC");
    wynik += na_cyfre(n % 10, "...");
    return wynik;
}
```

### wersja Python

```
def na_cyfre(cyfra, znaki):
    jednosc = znaki[0]
    piatek = znaki[1]
    dziesiatka = znaki[2]
    if cyfra == 4: return jednosc + piatek
    if cyfra == 9: return jednosc + dziesiatka
    wynik = ''
    if cyfra >= 5: wynik = piatek
    for i in range(cyfra % 5): wynik += jednosc
    return wynik

def na_rzymska(n):
    wynik = ''
    for i in range(n // 1000): wynik += 'M'
    wynik += na_cyfre(n % 1000 // 100, "CDM")
    wynik += na_cyfre(n % 100 // 10, "XLC")
    wynik += na_cyfre(n % 10, "...")
    return wynik
```

**Celem funkcji `na_rzymska` jest zamiana liczby arabskiej na jej zapis w systemie rzymskim. Jakim napisem należy uzupełnić trzy kropki w ciele funkcji `na_rzymska`, aby spełniała swoje zadanie?**

## 19. Rozważmy poniższą funkcję:

### wersja C++

```
int ile(int n, int m) {
    long long liczba = 1;
    for (int i = 1; i <= n; i++)
        liczba *= i;
    int wynik = 0;
    while (liczba % m == 0) {
        liczba /= m;
        wynik++;
    }
    return wynik;
}
```

### wersja Python

```
def ile(n, m):
    liczba = 1
    for i in range(1, n + 1):
        liczba *= i
    wynik = 0
    while liczba % m == 0:
        liczba //= m
        wynik += 1
    return wynik
```

**Jaki jest wynik wywołania `ile(18, 24)`?**

20. Rozważmy poniższe funkcje:

wersja C++

```
int iloczyn_cyfr(int n) {
    if (n == 0) return 1;
    return (n % 10) * iloczyn_cyfr(n / 10);
}

int ile_roznych(int n, int m) {
    set<int> wyniki;
    for (int i = 1; i <= n; i++) {
        int iloczyn = iloczyn_cyfr(i);
        if (iloczyn <= m)
            wyniki.insert(iloczyn);
    }
    return wyniki.size();
}
```

Jaki jest wynik wywołania `ile_roznych(1000000, 30)`?

wersja Python

```
def iloczyn_cyfr(n):
    if n == 0: return 1
    return (n % 10) * iloczyn_cyfr(n // 10)

def ile_roznych(n, m):
    wyniki = set()
    for i in range(1, n + 1):
        iloczyn = iloczyn_cyfr(i)
        if iloczyn <= m:
            wyniki.add(iloczyn)
    return len(wyniki)
```

