Rozwiązania testu wiedzy algorytmicznej

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

Olimpiada Informatyczna Juniorów

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?

5

Rozwiązanie: Gwiazdka zostanie wypisana dla $i \in \{0, 2, 4, 6, 8\}$. Gdy i osiągnie wartość 10, pętla zostanie przerwana.

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?

- ☐ `y >= x`
- ☒ `y >= z`
- ☐ `x >= z`
- ☐ `y < z`

Rozwiązanie: Funkcja najpierw sprawdza czy x jest maksimum (czy jest większe lub równe od y oraz z). Jeżeli kolejny warunek (ten, który należy uzupełnić) zostanie sprawdzony, oznacza to, że x nie jest maksimum, a więc, że wystarczy rozstrzygnąć między y oraz z. Jeżeli więc $y \geq z$, należy zwrócić y, a w przeciwnym przypadku z.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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.')
elif bmi <= 25.0:
  print('Waga w normie.')
elif bmi <= 30.0:
  print('Nadwaga.')
else:
  print('Otylosc.')
```

Jaki napis wypisze powyższy fragment programu?

- ☐ `Niedowaga.`
- ☒ `Waga w normie.`
- ☐ `Nadwaga.`
- ☐ `Otylosc.`

Rozwiązanie: Powyższy kod oblicza współczynnik BMI (*Body Mass Index*) według wzoru $BMI = \frac{masa}{wzrost^2}$. W tym przypadku $masa = 90$, a $wzrost = 2$, czyli $BMI = \frac{90}{2^2} = 22.5$. Program zatem wypisze napis `Waga w normie.`, sprawdzając najpierw czy $BMI < 20$ (nie), a później sprawdzając czy $BMI \leq 25$ (tak). Pozostałe warunki nie zostaną nawet sprawdzone.

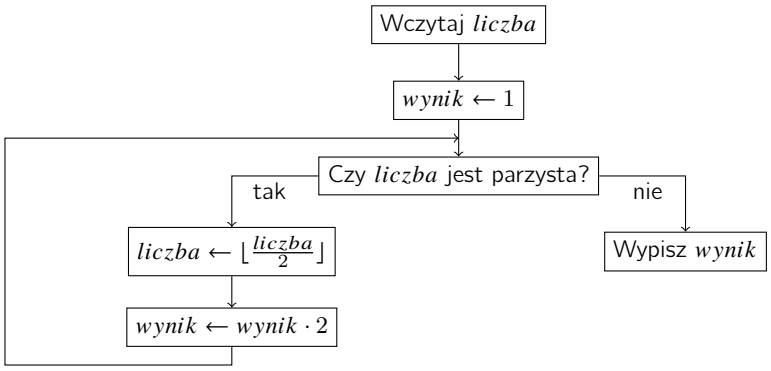
4. **Ciąg Fibonacciego zdefiniowany jest w następujący sposób: $F_0 = F_1 = 1$ oraz $F_n = F_{n-1} + F_{n-2}$ dla liczb naturalnych $n \geq 2$. Innymi słowy: zerowy i pierwszy wyraz ciągu to 1, zaś każdy kolejny wyraz jest sumą dwóch poprzednich. Kilka pierwszych wyrazów ciągu Fibonacciego to $1, 1,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

5

Rozwiązanie: Kolejne liczby Fibonacciego to: $13, 21, 34, 55, 89$. Następna (144) jest już trzycyfrowa, a ciąg jest rosnący. W ciągu Fibonacciego jest zatem dokładnie pięć liczb dwucyfrowych.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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: $\lfloor\frac{8}{3}\rfloor = 2$).**



4

Rozwiązanie: Schemat przedstawia algorytm znajdujący największą potęgę dwójki, która jest dzielnikiem liczby podanej na wejściu. Wczytana liczba systematycznie jest dzielona przez 2, a wynik mnożony przez 2, aż do momentu, w którym liczba jest już nieparzysta i wtedy wynik wypisywany jest na wyjście.

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 `potega(x, y)` **była równa x^y (x do potęgi y) dla wszystkich liczb naturalnych x, y spełniających warunek $1 \leq x, y \leq 9$?**

- ☐ `x * y`
- ☒ `x * potega(x, y - 1)`
- ☐ `potega(x, y)`
- ☐ `x ^ y`

Rozwiązanie: Aby obliczyć x^y można zastosować podejście rekurencyjne: $x^y = x \cdot x^{y-1}$ (y-krotne pomnożenie x przez siebie to x razy $(y-1)$-krotne pomnożenie x przez siebie).
Pozostałe odpowiedzi są błędne.
Pierwsza odpowiedź oblicza iloczyn liczb x i y, a nie potęgę.
Zastosowanie `potega(x, y)` prowadzi do nieskończonej rekursji w przypadku dodatnich y i zapętlenia programu.
Zapis `x ^ y` oznacza zaś operację XOR (zarówno w C++ jak i w Pythonie).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

7. **Rozważmy poniższą funkcję:**

wersja C++

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

wersja Python

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

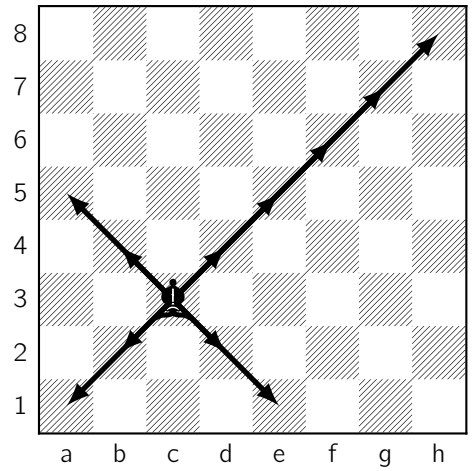
Ile jest nieujemnych liczb całkowitych n nie przekraczających 100, które podane jako argument funkcji `przeksztalcenie` **dają w wyniku wartość 4?**

3

Rozwiązanie:

Chodzi o liczby $n \in \{10, 11, 12\}$. Dla każdej z nich wartość w nawiasie jest większa lub równa 20, ale mniejsza niż 25. Po podzieleniu przez 5 i zaokrągleniu w dół otrzymamy 4.

8. **Rozważmy standardową szachownicę 8×8 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 goniec ustawiony na polu c3 atakuje 11 pól. Zakładamy, że goniec nie atakuje pola c3.

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

20

Rozwiązanie:

Goniec ustawiony na przykład na polu a1 atakuje zaledwie $A = 7$ pól. Jest to jednocześnie najmniejsza możliwa liczba pól, które może zaatakować.

Goniec ustawiony na jednym ze środkowych pól szachownicy (na przykład na polu d4) atakuje najwięcej pól, czyli $B = 13$.

Wartość $A + B$ wynosi zatem $7 + 13 = 20$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

9. **Rozważmy poniższą funkcję:**

wersja C++

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

wersja Python

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

Załóżmy następującą definicję:
(w C++)
`vector<int> pozycje({ 3, 0, 7, 6 });`
(w Pythonie)
`pozycje = [3, 0, 7, 6]`
Jaki będzie wynik wywołania `konstruuj("olimpiada", pozycje)`**?**

moda

Rozwiązanie:
Funkcja konstruuje napis korzystając z liter podanego słowa zgodnie z podaną listą pozycji. Należy pamiętać o tym, że (zarówno w C++ jak i w Pythonie) pozycje w napisach numerowane są od 0. A zatem znak numer 1 słowa `olimpiada` to litera `l`, znak numer 6 to litera `a` itd.

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.

- ☐ `zmieniaj(5, 10, 1)`
- ☒ `zmieniaj(8, 20, 2)`
- ☐ `zmieniaj(20, 8, 2)`
- ☒ `zmieniaj(3, 13, 5)`

Rozwiązanie:
Funkcja systematycznie zwiększa zmienną x o wartość δ (delta) oraz zmniejsza y o δ, aż zmienne staną się równe. Jeżeli w którymś momencie otrzymamy $x > y$, to już zawsze tak będzie i program się zapętli (z dokładnością do ewentualnego przepełnienia typu `int` w przypadku C++).
Na podstawie tego od razu możemy uznać odpowiedź `zmieniaj(20, 8, 2)` za błędną.
W przypadku `zmieniaj(5, 10, 1)` również program się nie zakończy, bo różnica $x - y$ będzie stale nieparzysta.
W pozostałych przypadkach funkcja zakończy się: `zmieniaj(8, 20, 2)` zakończy się na wartości 14 (po trzech iteracjach), zaś `zmieniaj(3, 13, 5)` zakończy się po zwróceniu 8 (po jednej iteracji).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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

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

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: $\lfloor \frac{8}{3} \rfloor = 2$).

32

Rozwiązanie:
Funkcja oblicza $\lfloor \log_2 n \rfloor$ (wartość logarytmu dwójkowego zaokrągloną w dół), czyli największą potęgę dwójki przez jaką możemy podzielić n aż osiągniemy wartość 1 (po zaokrągleniu wyniku w dół). Odpowiedzią jest więc $2^5 = 32$.
Możliwe jest również obliczenie odpowiedzi zauważając jedynie, że R jest niemalejąca i wyznaczając dla każdej wartości najmniejszy argument funkcji, który powoduje uzyskanie takiego wyniku:

- $R(2) = R(3) = 1$,
- $R(4) = R(5) = R(6) = R(7) = 2$,
- $R(8) = R(9) = \ldots = R(15) = 3$,
- $R(16) = R(17) = \ldots = R(31) = 4$,
- $R(32) = R(33) = \ldots = R(63) = 5$.

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 $a + b$.

7

Rozwiązanie: Jeżeli $f(n) = 2n + 1$, to $f(f(n)) = f(2n + 1) = 2(2n + 1) + 1 = 4n + 3$, stąd $a = 4$, $b = 3$, czyli $a + b = 7$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

13. **Rozważmy poniższy fragment programu:**

wersja C++

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

wersja Python

```
napis = 'ooolimpiadaaa'
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?

33

Rozwiązanie: Fragment programu sprawdza liczbę par indeksów (i, j), że i-ta litera jest równa j-tej literze napisu `olimpiadaaa`.
W napisie są cztery litery `a` ($4 \cdot 4 = 16$ możliwości wyboru pary (i, j)), trzy litery `o` ($3 \cdot 3 = 9$ możliwości wyboru pary), dwie litery `i` ($2 \cdot 2 = 4$ możliwości wyboru pary) oraz po jednej literze (i jednej możliwości wyboru pary) `d`, `l`, `m`, `p`.
Razem otrzymujemy wynik $16 + 9 + 4 + 1 + 1 + 1 + 1 = 33$.

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.

- ☒ `kajak`
- ☐ `ioi`
- ☒ `oij`
- ☐ `olimpiada`

Rozwiązanie: Funkcja tworzy dla każdej z 26 małych liter alfabetu angielskiego komórkę, która ustawiona będzie na `true`/`True`, gdy w napisie znajduje się chociaż jedno wystąpienie danej litery. Następnie zliczane są wartości `true`/`True` z tablicy, a więc funkcja zwraca liczbę różnych małych liter znajdujących się w napisie.
Napisy `kajak` oraz `oij` składają się z trzech różnych liter, zaś napisy z pozostałych wariantów odpowiedzi nie.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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

16

Rozwiązanie: Ponieważ $9 = 3^2$, każda cyfra z systemu dziewiątkowego tłumaczy się na dwie cyfry z systemu trójkowego. W istocie: wagi kolejnych cyfr (od prawej strony liczby) w systemie trójkowym to $3^0, 3^1, 3^2, 3^3, 3^4, \ldots$, czyli $1 \cdot 9^0, 3 \cdot 9^0, 1 \cdot 9^1, 3 \cdot 9^1, 1 \cdot 9^2, \ldots$.
Możliwe jest więc rozpatrywanie każdej cyfry liczby $8251325_{(9)}$ z osobna:

- $8_{(9)} = 22_{(3)}$,
- $2_{(9)} = 02_{(3)}$,
- $5_{(9)} = 12_{(3)}$,
- $1_{(9)} = 01_{(3)}$,
- $3_{(9)} = 10_{(3)}$,
- $2_{(9)} = 02_{(3)}$,
- $5_{(9)} = 12_{(3)}$,

A zatem $8251325_{(9)} = 22021201100212_{(3)}$, czyli suma cyfr to 16.

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 $s(s(n)) = 10$.**

199

Rozwiązanie: Liczby, które mają sumy cyfr 10 to: $19, 28, 37, \ldots$. Liczby o sumie cyfr 28 lub większej muszą być czterocyfrowe (każda cyfra może być równa co najwyżej 9). Sumę cyfr 19 można jednak uzyskać liczbą trzycyfrową na wiele różnych sposobów, w szczególności używając jako cyfry setek 1, co wymusza, aby pozostałe cyfry były równe 9. Otrzymamy w ten sposób liczbę 199, czyli rozwiązanie zadania, ponieważ wszystkie mniejsze liczby naturalne mają sumę cyfr równą co najwyżej 18.

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)`?

128

Rozwiązanie: Łatwo sprawdzić, że `wypisuj(1)` wypisuje jedną gwiazdkę.
Podobnie, `wypisuj(2)` wypisuje gwiazdkę oraz uruchamia `wypisuj(1)` wypisując kolejną gwiazdkę (razem dwie gwiazdki).
Analizując dalej `wypisuj(3)` wypisuje jedną gwiazdkę oraz uruchamia `wypisuj(2)` (wypisujące dwie gwiazdki) oraz `wypisuj(1)` (jeszcze jedna gwiazdka). Razem daje to 4 gwiazdki.
Ogólnie, możemy łatwo pokazać, że dla większych n funkcja `wypisuj` wypisuje dokładnie $2^{n-1} = 1 + 2^{n-2} + 2^{n-3} + \ldots + 2^0$ gwiazdek, a zatem dla $n = 8$ wypisanych zostanie $2^7 = 128$ gwiazdek.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

18. **Rozważmy poniższe funkcje:**

wersja C++

```
string na_cyfre(int cyfra, string znaki) {
  char jednosc = znaki[0];
  char piatka = znaki[1];
  char dziesiatka = znaki[2];
  if (cyfra == 4) return string(1, jednosc) + piatka;
  if (cyfra == 9) return string(1, jednosc) + dziesiatka;
  string wynik = "";
  if (cyfra >= 5) wynik = piatka;
  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]
  piatka = znaki[1]
  dziesiatka = znaki[2]
  if cyfra == 4: return jednosc + piatka
  if cyfra == 9: return jednosc + dziesiatka
  wynik = ''
  if cyfra >= 5: wynik = piatka
  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?**

IVX

Rozwiązanie: Funkcja `na_rzymska` rozpatruje kolejne cyfry zapisu dziesiętnego liczby n i skleja wyniki uzyskane ze skonwertowania pełnej liczby tysięcy, setek, dziesiątek i jedności w finalny wynik. W przypadku setek, dziesiątek i jedności jest za to odpowiedzialna funkcja `na_cyfre`, która dla każdej liczby setek, dziesiątek lub jedności (z zakresu 0 do 9) zwraca odpowiedni zapis.

Pierwszy znak napisu `znaki` oznacza symbol jednej setki/dziesiątki/jedności, drugi znak oznacza symbol pięciu, a trzeci – symbol dziesięciu. W systemie rzymskim jedna jedność to `I`, pięć jedności to `V`, zaś dziesięć jedności to `X` i taki właśnie napis należy wysłać do funkcji `na_cyfre`.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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)`**?**

5

Rozwiązanie: Funkcja najpierw oblicza zmienną `liczba`, która jest równa iloczynowi liczb od 1 do n włącznie, czyli wartość $n!$ (n silnia). Następnie liczona jest największa potęga liczby m, która jest dzielnikiem $n!$.
W przypadku wywołania `ile(18, 24)` najpierw obliczana jest wartość $18! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot 18$ i sprawdzane jest ile razy można tę liczbę podzielić bez reszty przez 24.
Liczba $24 = 2^3 \cdot 3$, a zatem w liczbie $18!$ interesuje nas liczba wystąpień czynnika pierwszego 2 oraz liczba wystąpień czynnika 3. Każde podzielenie przez 24 będzie bowiem „zużywać" z liczby $18!$ trzy czynniki 2 oraz jeden czynnik 3.
Liczba $n!$ zawiera w rozkładzie na czynniki pierwsze dokładnie $\lfloor \frac{n}{m} \rfloor + \lfloor \frac{n}{m^2} \rfloor + \lfloor \frac{n}{m^3} \rfloor + \ldots$ czynników pierwszych m. Istotnie, co m-ty czynnik iloczynu $1 \cdot 2 \cdot \ldots \cdot n$ ma w rozkładzie co najmniej jeden czynnik m, co (m^2)-ty czynnik tego iloczynu ma w rozkładzie co najmniej jeszcze jeden (wcześniej niepoliczony) czynnik m itd.
Ostatecznie otrzymujemy, że liczba $18!$ w rozkładzie na czynniki pierwsze ma $\lfloor \frac{18}{2} \rfloor + \lfloor \frac{18}{4} \rfloor + \lfloor \frac{18}{8} \rfloor + \lfloor \frac{18}{16} \rfloor = 9+4+2+1 = 16$ czynników 2 oraz $\lfloor \frac{18}{3} \rfloor + \lfloor \frac{18}{9} \rfloor = 6 + 2 = 8$ czynników 3.
Ponieważ $24^5 = (2^3 \cdot 3)^5 = 2^{15} \cdot 3^5$, zmienna `wynik` będzie równa co najmniej 5. A ponieważ $24^6 = (2^3 \cdot 3)^6 = 2^{18} \cdot 3^6$, to kolejnych dwójek zabraknie na kolejne podzielenie przez 24, a więc zmienna `wynik` będzie równa dokładnie 5.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

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();
}
```

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)
```

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

23

Rozwiązanie: Funkcja `ile_roznych(n, m)` oblicza iloczyny cyfr kolejnych liczb naturalnych od 1 do n i umieszcza w zbiorze (`set`) te iloczyny, które nie przekraczają m.

Pytanie postawione w tym zadaniu dotyczy zatem ustalenia liczby iloczynów nie przekraczających 30, wśród liczb naturalnych od 1 do miliona. Iloczyn cyfr dowolnej liczby może mieć w rozkładzie na czynniki pierwsze jedynie czynniki 2, 3, 5 lub 7. Istotnie, większe czynniki pierwsze wymagałyby istnienia cyfr większych niż 10, bo nie da się ich osiągnąć jako iloczyn kilku cyfr. W ten sposób łatwo pokazać, że iloczyny $11, 13, 17, 19, 23, 29$, a także $22 = 11 \cdot 2$ oraz $26 = 13 \cdot 2$, są nieosiągalne, niezależnie od zakresu n liczb jakie sprawdzamy.

Wszystkie pozostałe iloczyny od 0 do 30 włącznie (jest ich 31) są zaś możliwe do osiągnięcia. Dowolny iloczyn do 30 składa się z co najwyżej pięciu czynników pierwszych, a więc liczby co najwyżej pięciocyfrowe będą wystarczające. Wynikiem zadania jest więc 31 minus 8 niemożliwych do uzyskania iloczynów, czyli $31 - 8 = 23$.

W istocie, można pokazać nawet więcej: już `ile_roznych(56, 30)` równe jest 23 (i rzeczywiście liczba 56 jest najmniejszą liczbą potrzebną do uzyskania iloczynu 30, a wszystkie inne osiągalne iloczyny można zrobić z użyciem liczb mniejszych niż 56).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".