

Test wiedzy w wersji Python (z odpowiedziami)

XV OIJ, zawody I stopnia, tura testowa
7 października 2020



Poniżej znajdują się rozwiązania zadań pierwszego podejścia tury testowej zawodów I stopnia XV Olimpiady Informatycznej Juniorów (oij.edu.pl).

1. Która z poniższych instrukcji pozwala wczytać liczbę wprowadzoną z klawiatury i zapisać ją w zmiennej x typu `int`?

- `input x`
- `x = int(input())`
- `get_integer(&x)`
- `read(x)`

2. Celem poniższego (fragmentu) programu jest obliczyć sumę liczb naturalnych od 1 do n :

```
suma = 0
for i in range(1, n+1):
    ???
```

Co należy wstawić w miejsce znaków zapytania, aby program spełniał swoje założenia?

- `suma += 1`
- `suma = i`
- `suma += i`
- `suma = suma + 1`
- `suma + i`

3. Celem poniższego (fragmentu) programu jest obliczyć liczbę liter a w słowie S :

```
ile_a = 0
for i in range(len(S)):
    if ???:
        ile_a += 1
```

Co należy wstawić w miejsce znaków zapytania, aby program spełniał swoje założenia?

- `S == 'a'`
- `S[i] = 'a'`
- `i == 'a'`
- `S[i] == 'a'`
- `S == "a"`



4. Celem poniższego (fragmentu) programu jest znaleźć największą liczbę w tablicy pięcioelementowej:

```
maksimum = 0
for i in range(5):
    if tab[i] > maksimum:
        maksimum = tab[i]
print(maksimum)
```

Dla jakich danych wypisana odpowiedź nie będzie prawidłowa?

- [14, 0, 5, 6, -2]
- [0, -3, -2, 0, -5]
- [7, -1, 9, -23, 1]
- [3, 5, 1, 9, 2]
- [-5, -4, -9, -8, -3]

Rozwiązanie:

Błąd polega na nieprawidłowej początkowej wartości zmiennej maksimum. Jeśli wszystkie liczby są mniejsze od 0, zmienna ta pozostanie równa 0. Przykładowa poprawka mogłaby polegać na mniejszej wartości początkowej lub wręcz ustawieniu jej na pierwszy element tablicy tab.

5. Celem poniższego (fragmentu) programu jest wypisać na wyjście liczby 10, 9, 8, ..., 1 (w tej kolejności):

```
for i in range(10):
    print(???)
```

Co należy wstawić w miejsce znaków zapytania?

- 10 + i
- i -= 1
- 11 - i
- 10 - i
- i - 10
- 10

6. Rozważmy fragment programu pokazany poniżej:

```
for i in range(1, 6):
    for j in range(i+1, 6):
        print('*', end='')
```

Ile znaków * zostanie wypisanych przez powyższy kod?

Rozwiązanie:

Dla $i = 1$ gwiazdka zostanie wypisana dla $j \in \{2, 3, 4, 5\}$, dla $i = 2$ gwiazdka zostanie wypisana dla $j \in \{3, 4, 5\}$ i tak dalej. Razem: $4 + 3 + 2 + 1 = 10$ gwiazdek.

7. Na lewej szalce wagi szalkowej położono przedmiot o masie 1. Prawa szalka wagi jest aktualnie pusta. Dysponujesz dowolną liczbą odważników o masie 7 każdy oraz dowolną liczbą odważników o masie 11 każdy. Ile najmniej odważników należy położyć na wagę, żeby była w równowadze? Zakładamy, że odważniki można kłaść na obu szalkach wagi.

Rozwiązanie:

Rozwiązujemy równanie $7x + 11y = 1$ (w programie można to zrobić na przykład rozszerzonym algorytmem Euklidesa, a na kartce można rozwiązanie zgadnąć). Rozwiązanie $x = -3$, $y = 2$ daje najmniejszą wartość $|x| + |y|$ (liczbę użytych odważników). To oznacza, że wystarczy 5 odważników: trzy odważniki o masach 7 każdy kładziemy na lewą szalkę, a dwa odważniki o masach 11 każdy kładziemy na prawą szalkę.

8. Ile najwięcej tablic zmiennych typu `int` rozmiaru 1000×1000 każda można umieścić w programie, aby jego zużycie pamięci nie przekroczyło 100 MB?

- około dziesięciu
- około miliona
- żadnej
- jedną
- około tysiąca
- około stu

Rozwiązanie:

Jedna zmienna typu `long long int` zajmuje zwykle kilka bajtów (około 8 plus niewielki dodatkowy narzut na przechowywanie wskaźników do podtablic). Milion takich zmiennych zajmuje więc około 8 MB. A zatem w limicie 100 MB takich tablic zmieści się niewiele ponad dziesięć.

9. Które z poniższych wyrażeń są równe $\lceil \frac{x}{2} \rceil$ czyli połowie x zaokrąglonej w górę do najbliższej liczby całkowitej?

- $x // 2$
- $(x + 1) // 2$
- $x // 2 + 1$
- $x // 2 + x \% 2$
- $(3 * x + 2) // 6$

10. Jaki zbiór nominałów pozwala wydać każdą całkowitą kwotę w przedziale $[10, 100]$? Zakładamy, że każdego nominału można używać dowolnie wiele razy.

- {3, 6, 10}
- {10, 11, 12, 13}
- {2, 5}
- {1}
- {2, 3}

Rozwiązanie:

Dysponując nominałami {3, 6, 10} nie jest możliwe wydanie kwoty 11, a dysponując nominałami {10, 11, 12, 13} nie jest możliwe wydanie kwoty 14. W pozostałych przypadkach odpowiedź jest pozytywna: odpowiednią liczbą nominałów 1 da się wydać każdą całkowitą kwotę. Aby przekonać się, że odpowiedź dla zbiorów {2, 5} oraz {2, 3} należy wyznaczyć sposoby wydania kwot 10 oraz 11 tymi nominałami. Każdą większą kwotę można osiągnąć dokładając do jednej z nich odpowiednią liczbę nominałów 2.

11. Rozważmy fragment poniższego programu:

```
x = ???
if x != 2:
    if x % 3 == 2:
        print('tak', end='')
    if x % 5 == 2:
        print('owo', end='')
```

Jaką najmniejszą liczbę naturalną można wstawić w miejsce znaków zapytania, aby program wypisał napis takowo?

17

Rozwiązanie:

Liczby, które dają resztę 2 przy dzieleniu przez 3 i 5 to te same liczby, które dają resztę 2 przy dzieleniu przez 15: czyli 2, 17, 32, 47 i tak dalej. Liczba 2 nie jest rozwiązaniem (bo program sprawdza czy $x \neq 2$), więc najmniejszy x , który spowoduje wypisanie obu napisów to 17.

12. Które z poniższych liczb zapisanych w systemie dwójkowym są parzyste?

- 10001₂
- 10100₂
- 10011₂
- 10010₂
- 11100₂

Rozwiązanie:

W systemie binarnym parzyste są te liczby, które kończą się na 0.

13. Rozważmy fragment programu pokazany poniżej:

```
def generuj(n):
    if n == 0: return "a"
    return "b" + generuj(n - 1)
```

Jaki napis zwróci wywołanie generuj(3)?

- bba
- bbb
- baaa
- abbb
- bbba

14. Co można wstawić w miejsce znaków zapytania, żeby program wypisał napis kapusta?

```
if (2 + 2 == 4) ??? (True != False)
    print("kapusta")
```

- and
- ==
- not
- ^
- or

15. Rozważmy poniższą funkcję:

```
def f(x):  
    return x * x
```

Wywołanie $f(f(x))$ oblicza:

- x^2
- x
- x^3
- x^4

Rozwiązanie:

Skoro $f(x) = x^2$, to $f(f(x)) = f(x^2) = (x^2)^2 = x^4$.

16. Celem poniższego programu jest wypisać na wyjście kolejne litery alfabetu angielskiego (tzn. napis abcdefghijklmnopqrstuvwxyz):

```
znak = 'a'  
for i in range(26):  
    print(???, end='')
```

Co należy wstawić w miejsce znaków zapytania?

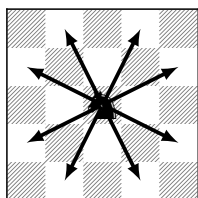
- `chr(ord(znak) + i)`
- `'a'`
- `chr(ord(znak) + 'a')`
- `znak + 'i'`
- `znak`

17. Rozważmy fragment programu pokazany poniżej:

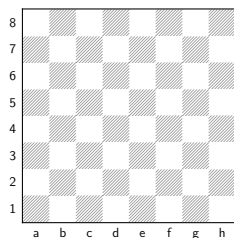
```
liczba = 3  
i = 1  
while i <= ???:  
    liczba = (3 * liczba + 1) // 2  
    i *= 2  
print(liczba)
```

Program po skompilowaniu i uruchomieniu wypisał na ekranie liczbę 27. Jaką najmniejszą liczbę całkowitą można wstawić w miejsce znaków zapytania, aby tak się stało?

18. Ile najwięcej skoczków szachowych można umieścić na szachownicy, aby żaden nie mógł wskoczyć na drugiego w jednym ruchu? Zakładamy, że na każdym polu można umieścić co najwyżej jednego skoczka.



Ruchy skoczka szachowego



Pola szachownicy

Rozwiązanie:

Skoczki atakuje pola o przeciwnym kolorze. Jeśli więc ustawimy wszystkie skoczki na polach białych lub wszystkie skoczki na polach czarnych, uda się zapełnić połowę szachownicy (czyli 32 pola).

Aby przekonać się, że nie jest możliwe uzyskanie lepszego wyniku, można sprawdzić sytuację dla szachownicy wymiaru 4×4 i przekonać się że da się tam ustawić maksymalnie osiem skoczków.

19. Rozważmy fragment programu pokazany poniżej:

```
suma = 0
for i in range(1, 247):
    suma += i % 10
print(suma)
```

Jaką liczbę wypisze powyższy program?**Rozwiązanie:**

Dodana wartość będzie taka sama dla wartości i dających tę samą resztę z dzielenia przez 10. A zatem dla $i \in \{1, 2, \dots, 10\}$ do sumy zostanie dodane tyle samo co dla $i \in \{11, 12, \dots, 20\}$ i tak dalej. A zatem wynik to: $24 \cdot (0 + 1 + 2 + \dots + 9) + (0 + 1 + 2 + \dots + 6) = 24 \cdot 45 + 21 = 1101$.

20. Rozważmy poniższą funkcję:

```
def wypisuj(n):
    if n == 0:
        print('*', end='')
        return
    wypisuj(n - 1)
    wypisuj(n - 1)
```

Ile znaków * zostanie wypisanych przez wywołanie wypisuj(6) i wszystkie podwywołania rekurencyjne?**Rozwiązanie:**

Wywołanie funkcji uruchamia dwa podwywołania z parametrem o 1 mniejszym. Rekursja kończy się wypisaniem pojedynczej gwiazdki, gdy parametr jest równy 0. A zatem liczba wypisanych gwiazdek to 2^n , czyli w naszym przypadku $2^6 = 64$.

21. Rozważmy słowa o długości 6 złożone jedynie z liter a, b oraz c. Ile spośród tych słów zawiera w sobie (co najmniej raz) spójny fragment abc?**Rozwiązanie:**

Jest $3^3 = 27$ słów, które mają abc na samym początku, 27 słów które mają abc od drugiej do czwartej litery, 27 słów z abc od trzeciej do piątej litery oraz 27 słów z abc od czwartej do szóstej litery. W ten sposób policzymy jednak słowo abcabc podwójnie. A zatem wynik to: $27 + 27 + 27 + 27 - 1 = 107$.

22. Rozważmy fragment programu pokazany poniżej:

```
def ile_iloczynow(n):
    zbior = set()
    for i in range(1, n+1):
        for j in range(1, n+1):
            zbior.add(i * j)
    return len(zbior)
```

Jaką wartość zwróci wywołanie `ile_iloczynow(5)`?

Rozwiązanie:

Pamiętając, że set usuwa powtórzenia (tzn. przechowuje jedynie jedną kopię tych samych wartości) wystarczy rozpisać tabliczkę mnożenia do 5 i policzyć ile jest tam różnych liczb.

23. Na ile sposobów można wybrać dokładnie trzy różne elementy ze zbioru $\{1, 2, 4, 8, 16, 26, 27, 28, 29, 30\}$, żeby otrzymać sumę 31? Sposoby różniące się jedynie kolejnością wybranych składników uznajemy za jednakowe.

Rozwiązanie:

Zauważamy, że każde rozwiązanie musi zawierać dokładnie jeden element ze zbioru $\{26, 27, 28, 29, 30\}$. Pozostaje nam odpowiednio $\{5, 4, 3, 2, 1\}$ do uzyskania jako suma dokładnie dwóch liczb ze zbioru $\{1, 2, 4, 8, 16\}$: tylko liczby 5 oraz 3 (mające dokładnie dwie jedyńki w zapisie dwójkowym) mogą być tak zapisane. Ostatecznie dostaniemy rozwiązania $1 + 2 + 28$ oraz $1 + 4 + 26$.

24. Rozważmy poniższą funkcję:

```
def funkcja(a, b):
    if b == 0: return 0
    h = funkcja(a, b // 2) * 2
    if b % 2 == 0: return h
    return h + a
```

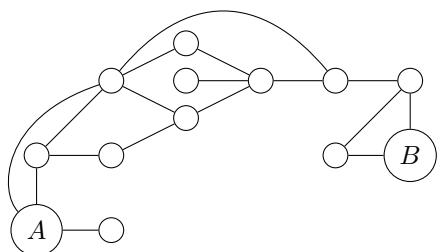
Co zwraca powyższa funkcja?

- a^b
- $a + b$
- $a \cdot b$
- $a \bmod b$ (reszta z dzielenia a przez b)

Rozwiązanie:

Przedstawiony algorytm to tak zwane mnożenie metodą chłopów rosyjskich. Algorytm przydaje się ze względu na właściwość, że w każdym kolejnym kroku wynik rośnie co najwyżej dwukrotnie. To zaś przydaje się na przykład w obliczeniach modulo liczba bliska zakresowi wartości typu liczbowego.

25. Ile krawędzi ma najdłuższa ścieżka między wierzchołkami A i B w grafie z rysunku? W tym zadaniu rozpatrujemy jedynie ścieżki, w których wszystkie odwiedzone wierzchołki muszą być parami różne.



Graf

10

26. Celem poniższej funkcji jest obliczyć wartość $\lfloor \sqrt{n} \rfloor$ (pierwiastek kwadratowy liczby n zaokrąglony w dół do najbliższej liczby całkowitej):

```
def pierwiastek(n):
    pocz, kon = 0, n
    while pocz < kon:
        srodek = (pocz + kon + 1) // 2
        if ???:
            pocz = srodek
        else:
            kon = srodek - 1
    return pocz
```

Co należy wstawić w miejsce znaków zapytania?

- $srodek * srodek < n$
- $srodek * srodek \geq n$
- $srodek * srodek \leq n$
- $pocz * kon \leq n$
- $pocz * pocz < kon * kon$

Rozwiązanie:

Przedstawiony kod implementuje znajdowanie pierwiastka metodą wyszukiwania binarnego. Wynik mieści się cały czas w przedziale pomiędzy $pocz$ i kon , obliczany jest $srodek$ (środek) tego przedziału (zaokrąglając w górę do całości). Jeśli $srodek$ podniesiony do kwadratu jest mniejszy lub równy n , to wiadomo, że wynik jest co najmniej równy $srodek$.

27. Celem poniższej funkcji jest obliczenie liczby najmniejszych elementów w niepustym ciągu liczb całkowitych:

```
def ile_minimow(liczby):
    ile, minimum = 0, liczby[0]
    for liczba in liczby:
        if liczba < minimum:
            minimum = liczba
            ???
        if liczba == minimum:
            ile += 1
    return ile
```

Jaką linię należy wstawić w miejsce znaków zapytania?

- liczba += 1
- ile = 1
- ile = 0
- continue
- nic nie trzeba wpisać (wystarczy zmazać znaki zapytania)

Rozwiązanie:

Jeżeli algorytm napotka na liczbę mniejszą niż bieżące minimum, liczba minimów powinna zostać wyzerowana. Zostanie ona zaraz po tym ustawiona na 1 wewnątrz warunku `liczba == minimum`.

28. Rozważmy program, który wczytuje liczbę naturalną n i wykonuje potem dokładnie $n^2 \log_2 n$ operacji elementarnych w celu obliczenia wyniku. Uruchamiono program na komputerze, który może wykonać 10^8 operacji elementarnych w ciągu sekundy. Dla jakiej wartości n wykonanie programu zajmie około dwie sekundy?

- 200
- 250 000
- 1 500
- 1 000 000
- 4 000

Rozwiązanie:

Wystarczy wiedzieć, że $2^{10} = 1024 \approx 1000$. A zatem \log_2 z wartości rzędu tysięcy jest około równy 10, a z wartości bliskich milionowi jest równy 20. Wynik więc powinien być około pierwiastka z 10^7 czyli pomiędzy 1000 i 10000. Po sprawdzeniu przekonamy się, że $4000^2 \cdot \log_2 4000$ to około 192 000 000.

29. Rozważmy fragment programu pokazany poniżej:

```
ile = 0
for liczba in range(1, 161):
    if liczba % 2 == 0:
        ile += 1
    elif liczba % 3 == 0:
        ile += 1
    elif liczba % 5 == 0:
        ile += 1
print(ile)
```

Jaką liczbę wypisze na wyjście ten program?

118

Rozwiązanie:

Pytanie schowane w powyższym zadaniu to ile jest liczb całkowitych od 1 do 160 podzielnych przez 2, 3 lub 5. Można to obliczyć z użyciem zasady włączeń i wyłączeń: jest $\lfloor \frac{160}{2} \rfloor = 80$ liczb podzielnych przez 2, $\lfloor \frac{160}{3} \rfloor = 53$ liczb podzielnych przez 3 oraz $\lfloor \frac{160}{5} \rfloor = 32$ liczb podzielnych przez 5. W ten sposób jednak policzyliśmy liczby podzielne przez 6, 10 lub 15 dwukrotnie, więc odejmujemy od wyniku $\lfloor \frac{160}{6} \rfloor = 26$, $\lfloor \frac{160}{10} \rfloor = 16$ oraz $\lfloor \frac{160}{15} \rfloor = 10$. Jednak wtedy nie policzymy liczb podzielnych przez 30 w ogóle i aby to naprawić, do wyniku należy dodać $\lfloor \frac{160}{30} \rfloor = 5$. A zatem wynik to: $80 + 53 + 32 - 26 - 16 - 10 + 5 = 118$.

30. Pewien program wykonuje $1 + 2 + 3 + 4 + \dots + \sqrt{n}$ operacji dla danej n . Jaką złożoność obliczeniową ma ten program?

- $\Theta(n \log n)$
- $\Theta(n^2)$
- $\Theta(n)$
- $\Theta(\log n)$
- $\Theta(\sqrt{n})$

Rozwiązanie:

Ponieważ jest $\frac{\sqrt{n}}{2}$ składników większych niż $\frac{\sqrt{n}}{2}$, suma jest większa niż $\frac{n}{4}$. A ponieważ jest \sqrt{n} składników, każdy nie przekraczający \sqrt{n} to suma jest co najwyżej n . To zaś oznacza, że złożoność obliczeniowa programu jest $\Theta(n)$.