

Test wiedzy wersja dla Pythona (z odpowiedziami)

XVI OIJ, zawody I stopnia, tura testowa
13 listopada 2021



1. Jakiego operatora należy użyć, aby sprawdzić czy zmienne `x` oraz `y` typu `int` zawierają tę samą zawartość?

- `eq`
- `==`
- `!==`
- `===`
- `=`

2. W jaki sposób zwiększyć wartość zmiennej `x` typu `int` o wartość zmiennej `y` typu `int`?

- `y + => x`
- `x = x + y`
- `x += y`
- `x = y`
- `x <- x + y`

3. W jaki sposób można obliczyć ostatnią cyfrę w zapisie dziesiętnym liczby naturalnej `a`?

- `a - 10 * (a // 10)`
- `a - (a // 10)`
- `a - (10 * a) // 10`
- `a % 10`
- `a // 10`

4. Rozważmy następujący fragment programu:

```
czy = True
for x in napis:
    if x != 'a':
        czy = False
```

Co zawiera zmienna `czy` po wykonaniu powyższego kodu?

- informację, czy `napis` jest niepusty
- zawsze wartość `False`
- liczbę liter w zmiennej `napis`
- liczbę liter innych niż `a` w zmiennej `napis`
- informację, czy `napis` jest pusty lub składa się jedynie z liter `a`

Rozwiązanie:

W pętli przeglądane są kolejne znaki napisu i zmienna `czy` zostanie ustawiona na `false` za pierwszym razem, w którym zostanie przeanalizowany znak inny niż litera `a`.



5. Które z poniższych wyrażeń w programie obliczają trzykrotność liczby zapisanej w zmiennej `x` typu `float`?

- `3x`
- `x + x + x`
- `x 3`
- `x / 2 * 6`
- `* x 3`
- `3 * x`

6. Rozważmy poniższą funkcję:

```
def oblicz(n):  
    wynik = 0  
    for i in range(n + 1):  
        wynik += i  
    return wynik
```

Jaka jest całkowita dodatnia wartość parametru `n`, żeby `oblicz(n) == 55`?

Rozwiązanie:

Funkcja oblicza sumę $1 + 2 + \dots + n$. Suma ta jest równa $\frac{n \cdot (n+1)}{2}$ co dla $n = 10$ jest równe 55.

7. Ile gwiazdek wypisze poniższy fragment programu?

```
for i in range(0, 11, 2):  
    print('*', end='')
```

Rozwiązanie:

Gwiazdka zostanie wypisana dla $i \in \{0, 2, 4, 6, 8, 10\}$.

8. Rozważmy poniższą funkcję:

```
def funkcja(napis):  
    wynik = ''  
    for i in range(len(napis) - 1, -1, -1):  
        wynik += chr(i + ord('a'))  
    return wynik
```

Jaki będzie wynik wywołania funkcji `funkcja('oij')`?

Rozwiązanie:

Funkcja dla każdego znaku napisu umieszcza znak o kodzie ASCII takim jak kod litery `a` plus wartość zmiennej `i`.

9. Rozważmy poniższą funkcję:

```
def funkcja(a, b):  
    wynik = 1  
    while b > 0:  
        wynik *= a  
        b -= 1  
    return wynik
```

Zakładając, że zmienne a oraz b są całkowite, dodatnie i nie przekraczają 9, które z poniższych wyrażeń jest równoważne wywołaniu funkcja(a , b)?

- a^b
- $\frac{a}{b}$
- $a + b$
- $a - b$
- $a \cdot b$

Rozwiązanie:

Funkcja w zmiennej wynik umieści wartość $\underbrace{a \cdot a \cdot \dots \cdot a}_b$ czyli a^b .

10. Ile równy jest największy wspólny dzielnik liczb 10^6 i 6^{10} ? Odpowiedź podaj jako liczba w systemie dziesiętkowym.

64

Rozwiązanie:

$10^6 = (2 \cdot 5)^6 = 2^6 \cdot 5^6$. Stąd największy wspólny dzielnik podanych liczb jest równy $2^6 = 64$.

11. Rozważmy poniższą funkcję:

```
def slow_phi(n):  
    wynik = 0  
    for i in range(1, n + 1):  
        if gcd(n, i) == 1:  
            wynik += 1  
    return wynik
```

Założmy, że wywołanie $\text{gcd}(x, y)$ zwraca największy wspólny dzielnik liczb x oraz y . Jaki będzie wynik wywołania $\text{slow_phi}(n)$ dla parametru $n = 10^9$? Odpowiedź podaj jako liczba w systemie dziesiętkowym.

400000000

Rozwiązanie:

Funkcja oblicza wartość funkcji ϕ Eulera, czyli liczbę liczb względnie pierwszych z n . Istnieje wzór na obliczenie wartości funkcji na podstawie rozkładu na czynniki pierwsze. Możliwe jest również uruchomienie powyższego programu (po własnoręcznej implementacji algorytmu Euklidesa na obliczanie NWD), wtedy wykona się w rozsądnym czasie (choć nieco dłuższym niż kilka sekund).

12. Rozważmy poniższą funkcję:

```
def jlt(n):
    wartosc = 1
    i = 1
    while True:
        wartosc += i * i * i
        if wartosc > n:
            return i - 1
        i += 1
```

Jaka jest największa wartość parametru n , dla której $jlt(n) == 3$?

13. Rozważmy poniższą funkcję:

```
def mx(a, b):
    wynik = 0
    for x in range(a, b + 1):
        y = x + 1
        wynik = max(wynik, x ^ y)
    return wynik
```

Jaką wartość zwróci wywołanie $mx(10, 45)$?

Rozwiązanie:

Funkcja oblicza największy XOR par liczb $(x, x+1)$ dla $x \in \{10, 11, \dots, 44\}$. Największy XOR będzie dla pary $(31, 32)$ ponieważ $31 = 11111_2$ zaś $32 = 10000_2$.

14. Rozważmy poniższą funkcję:

```
def cplp(tab, p):
    if p >= len(tab):
        return True
    if tab[p] % 2 == 0:
        return not cplp(tab, p + 1)
    return cplp(tab, p + 1)
```

Dla jakich zawartości tab wywołanie $cplp(tab, 0)$ zwróci True?

- [4, 9, 4, 1, 4]
- [2, 12, 22, 32]
- [2, 3, 4, 5]
- [7, 1, 2, 3]
- [5, 9, 13, 19]
- [2]

Rozwiązanie:

Funkcja sprawdza czy w tab znajduje się parzysta liczba elementów parzystych.

15. Rozważmy poniższą funkcję:

```
def czy_palindrom(slowo):  
    for i in range(len(slowo) // 2):  
        if slowo[i] != slowo[len(slowo) - 1 - i]:  
            ...  
    return True
```

Celem funkcji jest sprawdzenie, czy słowo podane jako parametr jest palindromem (czy jest takie samo niezależnie czy czytane jest od lewej czy od prawej). Jak należy uzupełnić kod w ..., aby to osiągnąć?

- return True
- i -= 1
- i = len(slowo)
- i += 1
- return False

Rozwiązanie:

Funkcja sprawdza pary znaków pierwszy-ostatni, drugi-przedostatni itd. Gdy znaki są różne, funkcja powinna zwrócić fałsz.

16. Ile dzielników większych niż 25 ma liczba 2 560?

12

Rozwiązanie:

$2560 = 256 \cdot 10 = 2^8 \cdot 2 \cdot 5 = 2^9 \cdot 5$. Dowolny dzielnik tej liczby jest postaci $2^\alpha \cdot 5^\beta$, gdzie $\alpha \in \{0, 1, \dots, 9\}$ oraz $\beta \in \{0, 1\}$. Każda kombinacja (α, β) prowadzi do uzyskania innego dzielnika, jest więc 20 różnych dzielników. Spośród nich 1, 2, 4, 5, 8, 10, 16, 20 są mniejsze lub równe 25.

17. Rozważmy algorytm sortowania bąbelkowego jak pokazany poniżej:

```
def sortuj(tab):  
    for i in range(1, len(tab)):  
        for j in range(1, len(tab)):  
            if tab[j] < tab[j - 1]:  
                tab[j - 1], tab[j] = tab[j], tab[j - 1]
```

Jak dużą tablicę tab można posortować z użyciem powyższego algorytmu, aby czas sortowania tą metodą na przeciętnym komputerze z roku 2020 był równy około pół sekundy?

- około 100 elementów
- około 2 000 elementów
- około 20 elementów
- około 100 000 elementów

Rozwiązanie:

Algorytmy $O(n^2)$ mają szansę (przy bardzo niewielkiej stałej) działać (w przypadku Pythona) nawet dla $n \approx 2000$ w czasie rzędu pół sekundy.

18. Ile jest liczb naturalnych n mniejszych niż 1000, których zapis w systemie rzymskim ma dokładnie 10 znaków?
Odpowiedź podaj w systemie dziesiętkowym.

24

Rozwiązanie:

Rozważmy każdą cyfrę arabską z osobna. Jedynie cyfra 8 powoduje uzyskanie czterech znaków rzymskich, cyfry 3 oraz 7 uzyskują trzy znaki rzymskie, a cyfry 2, 4, 6, 9 uzyskują dwa znaki rzymskie. Jeśli trzycyfrowa liczba arabska ma przetłumaczyć się na dziesięć znaków rzymskich to kolejne cyfry mają wnosić $4+4+2$ lub $4+2+4$ lub $2+4+4$ lub $4+3+3$ lub $3+4+3$ lub $3+3+4$ znaki rzymskie. Każda z powyższych możliwości przynosi 4 unikalne liczby arabskie spełniające warunki zadania.

19. Na przyjęcie przyszło około milion dzieci, każde z nich ma na koszulce zapisane inną liczbę naturalną od 1 do 10^{12} . Około milion razy będziemy potrzebowali sprawdzić czy na przyjęciu jest dziecko, które ma koszulkę z konkretną liczbą. Jaka struktura danych będzie najrozsądniejsza do tego zastosowania?

- dwie liczby naturalne: jedna która pamięta sumę liczb na koszulkach dzieci obecnych na przyjęciu, a druga będąca iloczynem liczb na koszulkach
- lista wartości typu `int` przechowująca numery na koszulkach w losowej kolejności
- lista wartości typu `bool`, w której na pozycji i przechowamy `True` wtedy i tylko wtedy, gdy na przyjęciu jest dziecko z koszulką z liczbą i
- set przechowujący numery na koszulkach dzieci obecnych na przyjęciu

Rozwiązanie:

Z sumy i iloczynu nie jest możliwe sprawdzanie kto jest na przyjęciu. Tablica `bool` zajęłaby zbyt dużo pamięci, a sprawdzanie czy konkretny element istnieje na `vectorze` zajmuje czas liniowy względem jego długości, czyli dla długości rzędu milion nie jest możliwe wykonanie milion takich sprawdzeń w rozsądnym czasie. Rozwiązanie z `unordered_set<int>` ma liniową złożoność czasową i pamięciową i jest zdecydowanie najlepsze z proponowanych.

20. Rozważmy poniższy fragment programu:

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

Jaką liczbę gwiazdek wypisaną przez program można osiągnąć, odpowiednio dobierając n ?

- 125
- 64
- 256
- 169
- 100
- 1000

Rozwiązanie:

Fragment programu wypisuje n^3 gwiazdek. A zatem poprawne odpowiedzi to sześćdziesiąt liczb naturalnych.

21. Rozważmy poniższą funkcję:

```
def wykonuj(n):  
    if n == 0:  
        print('*', end='')  
        return  
    wykonuj(n // 2)  
    wykonuj(n // 2)
```

Zakładamy, że operacje arytmetyczne oraz porównania na zmiennych typu `int` zajmują czas stały. Jaka jest pesymistyczna złożoność obliczeniowa funkcji `wykonuj`?

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

Rozwiązanie:

Funkcja wypisuje $2^{\lfloor \log_2 n \rfloor}$ gwiazdek. Wartość ta jest zawsze w przedziale $[\frac{n}{2}, n]$, stąd liczba wypisanych gwiazdek jest liniowa względem n . Koszt pozostałych operacji (sprawdzenia warunków, wywołania funkcji rekurencyjnych) można przypisać wypisywanym gwiazdkom, tak żeby wypisanie każdej gwiazdki odpowiadało za stałą liczbę innych operacji, stąd cała złożoność algorytmu jest liniowa.

22. Na ile sposobów można wybrać niektóre elementy ze zbioru $\{50, 10, 11, 40, 8, 9, 1, 2\}$, aby uzyskać sumę 60? Każdą liczbę można wybrać co najwyżej raz. Sposoby uznajemy za różne, jeżeli zbiory wybranych elementów są różne.

8

Rozwiązanie:

Sposoby te muszą zawierać albo element 50 i sumę 10 uzyskaną z pozostałych elementów (10 lub 9 + 1 lub 8 + 2) albo element 40 i sumę 20 uzyskaną z pozostałych elementów (11 + 9 lub 11 + 8 + 1 lub 10 + 9 + 1 lub 10 + 8 + 2 lub 9 + 8 + 2 + 1).

23. Która z poniższych liczb ma dokładnie dwie jedynki w zapisie dwójkowym?

- FF_{16}
- 20
- 3
- 100
- 1001_{16}
- 256



24. Dla jakich wartości parametru n poniższa funkcja poprawnie kończy swoje działanie (tzn. nie powoduje błędu wykonania)?

```
def wykonuj(n):  
    if n == 1:  
        return  
    wykonuj(100 % n)
```

- 13
 20
 9
 12

Rozwiązanie:

Problemy, które mogą tu potencjalnie wystąpić jest dzielenie przez 0 lub zapętlenie rekursji. Dla danej 20 następuje wywołanie `wykonuj(0)`, które prowadzi do dzielenia przez 0. Dla danej 12 następują wywołania `wykonuj(4)` oraz `wykonuj(0)`. Dla danych 9 oraz 13 ciąg wywołań kończy się przy uzyskaniu $n = 1$ bez żadnego błędu.

25. Ile jest napisów o długości 6 składających się jedynie z liter a oraz b, w których liczba liter a jest większa niż liczba liter b?

22

Rozwiązanie:

Jest jeden napis składający się z samych liter a, sześć napisów składających się z dokładnie jednej litery b oraz $\binom{6}{2} = 15$ napisów składających się z dwóch liter b.

26. Ile pamięci zajmuje tablica w poniższym kodzie?

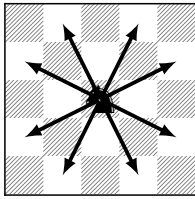
```
tab = list(range(1000000))
```

- około 20 GB
 około 500 MB
 około 900 MB
 około 400 kB
 około 36 MB

Rozwiązanie:

W Pythonie lista to ciąg wskaźników i obiektów z nimi związanych. Same wskaźniki zajmują ok. 8 MB, a pojedyncza (mała) liczba zajmuje zazwyczaj ok. 28 bajtów, czyli milion takich liczb zajmuje ok. 28 MB. Razem około 36 MB.

27. Ile pól może atakować skoczek na standardowej szachownicy 8×8 ? Zauważ, że skoczek nie atakuje pól leżących poza szachownicą, a więc dla pewnych ustawień skoczka na szachownicy może on atakować mniej niż 8 pól.



Kierunki ataku skoczka szachowego.

- 3
- 4
- 6
- 5
- 7
- 2

28. Rozważmy następującą funkcję:

```
def oblicz(operacje):  
    wynik = 0  
    for x in operacje:  
        if x == '+':  
            wynik += 1  
        if x == '*':  
            wynik *= 4  
    return wynik
```

Podaj napis złożony z czterech znaków + oraz dwóch znaków *, który należy przekazać jako parametr operacje, aby `oblicz(operacje) == 52`?

29. Rozważmy poniższą funkcję:

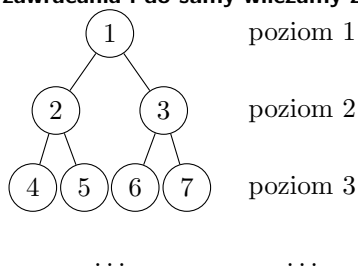
```
def oblicz(n, k):  
    wynik = 1  
    for i in range(1, n + 1):  
        wynik *= i  
    for i in range(1, k + 1):  
        wynik //= i  
    for i in range(1, n - k + 1):  
        wynik //= i  
    return wynik
```

Dla jakiego N wynikiem wywołania `oblicz(N, 3)` jest 20?

Rozwiązanie:

Funkcja oblicza $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$. Wartość $\binom{6}{3}$ jest równa 20.

30. Na poniższym rysunku wierzchołki są na trzech poziomach, każdy wierzchołek na poziomie innym niż ostatni jest połączony z dwoma na poziomie kolejnym. Wierzchołki są numerowane od 1 kolejno poziomami, a na każdym poziomie od lewej do prawej. Rozważmy podobny rysunek, ale o sześciu poziomach. Ile wynosi suma wartości wierzchołków na ciągu kolejno połączonych wierzchołków między wierzchołkami 35 oraz 45? Zakładamy, że poruszamy się bez zawracania i do sumy wliczamy zarówno 35 jak i 45.



Fragment pełnego drzewa binarnego (do trzech poziomów).

149

Rozwiązanie:

Ścieżka pomiędzy wierzchołkami to: $35 \rightarrow 17 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 11 \rightarrow 22 \rightarrow 45$.

