

## **Spis treści**

Zadanie 1A, 2023-11-16 .....	1
Zadanie 1B, 2023-11-16 .....	2
Zadanie 2A, 2024-01-04 .....	3
Zadanie 2B, 2024-01-04 .....	4
Zadanie 3A, 2024-01-18 .....	5
Zadanie 3B, 2024-01-18 .....	6
Bonus: Zasady oceniania 3A .....	7
Bonus: Zasady oceniania 3B .....	7

## **Zadanie 1A, 2023-11-16**

Dwie liczby naturalne większe od 1 są zgodne jeżeli dzielą się przez te same liczby pierwsze. Przykładem zgodnych liczb są pary: (6, 24), (40, 50), (13, 169), (44, 242), nie są zgodne np. pary: (6, 8), (40, 60), (13, 39), (44, 99).

Tablica  $T$  została wypełniona liczbami z zakresu [2..999]. Sąsiadami pola o indeksie  $i$  w tablicy są pola o indeksie  $j$ , gdy  $\text{abs}(i - j) < 3$ . Proszę napisać funkcję  $\text{zgodne}(T)$ , która dla tak wypełnionej tablicy zwraca liczbę elementów mających przynajmniej jednego zgodnego sąsiada. Po wykonaniu funkcji tablica nie musi pozostać nie zmieniona. Na przykład dla tablicy:  $T = [2, 3, 4, 5, 7, 6, 23, 24, 12, 13, 14, 15, 16, 45]$  funkcja powinna zwrócić wartość 7 (są to liczby 2, 4, 6, 24, 12, 15, 45).

Uwagi:

- Czas na rozwiązywanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.

## Zadanie 1B, 2023-11-16

Sudoku składa się z kwadratu o wymiarach  $9 \times 9$  podzielonego na 9 małych kwadratów o wymiarach  $3 \times 3$ . Na potrzeby zadania, kolejne małe kwadraty numerujemy wierszami od 1 do 9. W poprawnym rozwiążaniu w każdym wierszu, każdej kolumnie, i każdym małym kwadracie znajdują się liczby 1 – 9. W poprawnym rozwiążaniu umieszczonym w tablicy  $T$  ktoś zamienił miejscami dwa małe kwadraty. Proszę napisać funkcję  $sudoku(T)$ , która zwraca numery zamienionych kwadratów.

Dla poniższych danych wejściowych, poprawną odpowiedzą jest krotka (5, 9).

```
8 1 2 7 5 3 6 4 9  
9 4 3 6 8 2 1 7 5  
6 7 5 4 9 1 2 8 3  
1 5 4 3 6 8 8 9 6  
3 6 9 9 1 7 7 2 1  
2 8 7 4 5 2 5 3 4  
5 2 1 9 7 4 2 3 7  
4 3 8 5 2 6 8 4 5  
7 9 6 3 1 8 1 6 9
```

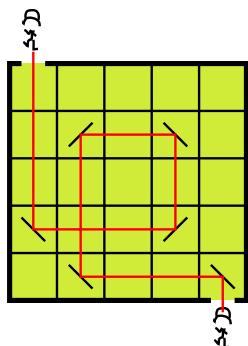
Uwagi:

- Czas na rozwiązywanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.

[pierwsze zadanie z trzeciego etapu olimpiady o diamentowy indeks 2022/2023]

## Zadanie 2A, 2024-01-04

Ogród składa się z  $N^2$  jednakowych kwadratowych działek. Posiada on dwa wejścia, jedno w lewym górnym rogu od strony północnej, drugie w prawym dolnym rogu od strony południowej. Dokładnie na środkach niektórych działek ustawiono obustronne lustra pod kątem  $45^\circ$  albo  $135^\circ$  tak, że patrzący przez wejście północne widzi osobę stojącą przy wejściu południowym.



Do ogrodu przyszedł zły człowiek i przekręcił dwa lustra, każde o  $90^\circ$ . Proszę napisać funkcję *napraw(ogrod)* poprawiającą położenie luster, tak aby przywrócić widoczność pomiędzy oboma wejściami. Ogród jest reprezentowany jako dwuwymiarowa tablica wypełniona spacjami. Lustra reprezentowane są odpowiednio jako znaki: / i \.

Uwagi:

- Czas na rozwiązywanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.

## Zadanie 2B, 2024-01-04

Na liczbach naturalnych możemy wykonywać następujące operacje:

1.  $A(n)$  zamienia liczbę  $n$  na sumę jej podzielników właściwych (mniejszych od samej liczby), np.  
 $A(1) = 1$ ,  $A(6) = 6$ ,  $A(12) = 16$ ,  $A(17) = 1$ .
2.  $B(n)$  zamienia liczbę  $n$  na najmniejszy, większy od tej liczby wyraz ciągu Fibonacciego, np.  
 $B(1) = 2$ ,  $B(4) = 5$ ,  $B(8) = 13$ .
3.  $C(n)$  zwiększa liczbę  $n$  o liczbę będącą rewersem liczby  $n$ , np.  $C(1) = 2$ ,  $C(10) = 11$ ,  $C(13) = 44$

Proszę napisać funkcję  $cycle(x,n)$ , która sprawdza czy startując od liczby  $x$  możemy do niej powrócić wykonując sekwencję operacji spośród A,B,C o długości większej od 1 i nie większej od  $n$ . Jeżeli jest to możliwe, funkcja powinna zwrócić długość znalezionej sekwencji operacji, w przeciwnym wypadku należy zwrócić wartość 0.

Na przykład wywołanie:

$cycle(29,6)$  powinno zwrócić 4 (cykl 29, B, 55, B, 89, C, 187, A, 29), [przykład jest błędny,  $B(29) = 34$ ]  
 $cycle(31,6)$  powinno zwrócić 0.

Uwagi:

- Czas na rozwiązywanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.

## Zadanie 3A, 2024-01-18

Dany jest ciąg  $N$  liczb naturalnych, z których wybieramy spójny fragment o długości  $K$  ( $1 < K < N$ ). Pomiędzy wszystkie elementy wybranego fragmentu możemy wstawiać operatory dodawania albo mnożenia, tak aby powstało wyrażenie arytmetyczne. W powstałym wyrażeniu nie mogą wystąpić dwa jednakowe operatory obok siebie. Interesuje nas znalezienie takiego fragmentu ciągu, który pozwala zbudować wyrażenie o wartości będącej liczbą pierwszą, taką że stosunek tej liczby pierwszej do długości znalezionej fragmentu jest największy. Proszę napisać funkcję `find_max(T)`, która dla ciągu zawartego w tablicy  $T$ , wyznaczy wartość maksymalnego ilorazu jaki można znaleźć. Jeżeli taki podciąg nie istnieje funkcja powinna zwrócić wartość zero.

Na przykład dla ciągu: 7, 8, 6, 4, 7, 3 funkcja powinna zwrócić wartość 16.6.

Mögliwe podciągi dające liczby pierwsze to:

$$7 + 8 \cdot 6 + 4 = 59, \quad 59/4 = 14.75$$

$$7 + 8 \cdot 6 + 4 * 7 = 83, \quad 83/5 = 16.6$$

$$6 \cdot 4 + 7 = 31, \quad 31/3 = 10.(3)$$

$$4 + 7 = 11, \quad 11/2 = 5.5$$

Uwagi:

- Czas na rozwiązanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.
- Nie wolno używać wbudowanej funkcji `eval`, można stosować mechanizm slicing-u.

## Zadanie 3B, 2024-01-18

Dana jest liczba odsyłaczowa, której elementy przechowują niepuste napisy składające się z małych liter alfabetu angielskiego. Proszę napisać funkcję `make_order(p)`, która porządkuje elementy listy tak, aby na jej początku znalazły się napisy, w których kolejne litery są w porządku rosnącym, natomiast na końcu listy znalazły się napisy, w których litery są w porządku malejącym. Pomiędzy powstałymi grupami elementów należy wstawić elementy zawierające pusty napis. Do funkcji należy przekazać wskaźnik do pierwszego elementu listy, funkcja powinna zwrócić wskazanie do uporządkowanej listy.

Na przykład lista: *ala* → *ola* → *abel* → *ula* → *irys* → *ewa* → *sroka* → *gips*

Po uporządkowaniu może mieć postać: *abel* → *gips* → „” → *irys* → *ala* → *ewa* → „” → *sroka* → *ola* → *ula*

Uwagi:

- Czas na rozwiązywanie zadania: 25 min.
- Oceniane będą: czytelność (komentarze), poprawność, efektywność rozwiązań.

Dodatkowe uwagi Garka podczas kolosa:

- Nie można tworzyć nowych elementów poza wartownikiem (którego trzeba usunąć) i dwoma węzłami z pustymi stringami.
- Można założyć, że mamy już zdefiniowane `class Node`.
- Lista nie zawiera napisów składających się z jednej litery.

## Bonus: Zasady oceniania 3A

<b>Podział punktów</b>	
komentarze	0.5
prime	1
eval	2
główna funkcja	1.5
<b>Błędy</b>	
brak komentarzy	-0.5
cały zakres w prime	-1
błąd w prime	-1
bez kolejności działań	-1
zupełnie zły wynik - wyrażenia	-1.5
brak dekompozycji (wszystko w jednej funkcji)	-0.2
sprawdzanie całego ciągu	-0.2
gubienie niektórych fragmentów	-0.2
dodatkowa tablica/lista	-0.5
nietrywialne wyjście poza zakres tablicy	-1

## Bonus: Zasady oceniania 3B

<b>Podział punktów</b>	
komentarze	0.5
funkcja sprawdzająca kolejność liter	2
główna funkcja	2.5
<b>Błędy</b>	
brak komentarzy	-0.5
tworzenie (alokowanie) nowych elementów	-2
zapomnienie wstawiania separatorów	-0.5
brak obsługi przypadku pustego łańcucha	-0.5
wychodzenie poza zakres tablicy/stringa	-1
uznanie ciągu stałego za silnie rosnący i malejący	-0.5
trzykrotne przejście po liście wejściowej	-0.5
dodatkowe szukanie końców łańcuchów	-0.5