

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Klocki –LOGIA 20 (2019/20), etap 2

Treść zadania

Tola układa z kwadratowych klocków spiralę na planszy podzielonej na cztery ćwiartki jak na rysunku obok. Pierwszy klocek umieszcza w pierwszej ćwiartce. Kolejne klocki układa spiralnie zgodnie z ruchem wskazówek zegara tak, jak na rysunku pomocniczym. Następnie próbuje policzyć, w której ćwiartce umieści ostatni klocek. Pomóż Toli rozwiązać problem.

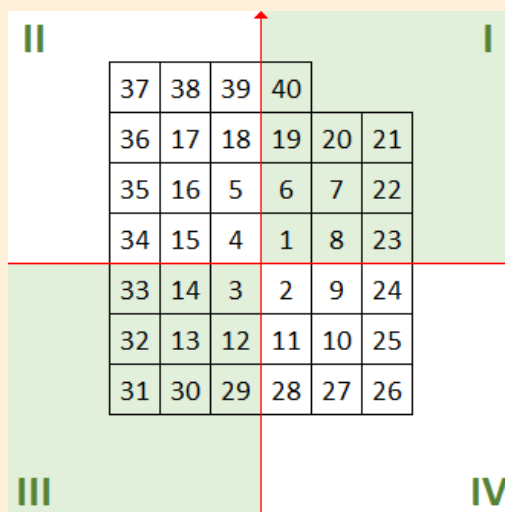
Napisz funkcję **klocki(n)**, której parametrem jest liczba klocków nie mniejsza niż 1 i nie większa niż 10 000. Wynikiem funkcji jest numer ćwiartki układu współrzędnych zapisany liczbą rzymską, w której Tola umieści ostatni klocek.

Przykłady:

Wynikiem **klocki(2)** jest 'IV'.

Wynikiem **klocki(17)** jest 'II'.

Wynikiem **klocki(40)** jest 'I'.

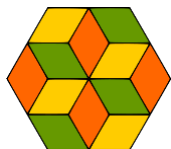


Omówienie rozwiązania

Zadanie Klocki można potraktować jako problem symulacyjny.

Przeanalizujmy sposób układania klocków dla początkowych wartości **n**.

numer ćwiartki	numer ostatniego klocka umieszczonego w danej ćwiartce		
	I cykl	II cykl	III cykl
I	1	$5 + 3 = 8$	$18 + 5 = 23$
IV	$1 + 1 = 2$	$8 + 3 = 11$	$23 + 5 = 28$
III	$2 + 1 = 3$	$11 + 3 = 14$	$28 + 5 = 33$
II	$3 + 2 = 5$	$14 + 4 = 18$	$33 + 6 = 39$



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Z powyższej analizy wynikają następujące wnioski:

- z każdym kolejnym cyklem rośnie liczba klocków dokładanych do danej ćwiartki,
- w ćwiartce II liczba dodanych klocków jest o jeden większa od liczby klocków, które zostały dodane w poprzedniej ćwiartce (III) – wyróżnienie kolorem czerwonym w tabeli (na przykład I cykl),
- podobna reguła dotyczy ćwiartki I – wyróżnienie kolorem niebieskim w tabeli (na przykład przejście z II do III cyklu).

Na podstawie obserwacji możemy ustalić schemat postępowania umożliwiający nam wyznaczenie numeru ćwiartki, w której znajdzie się dany klocek:

```
numer = 0
ile_d = 0
dopóki numer < n
    jeśli cwiartka == "I" lub cwiartka == "II"
        ile_d ← ile_d + 1
    numer = numer + ile_d
    jeśli numer >= n
        wynik cwiartka
```

W pojedynczym obrocie pętli ustalamy numer ostatniego klocka umieszczonego w danej ćwiartce. Jeśli numer ten jest większy lub równy liczbie klocków do ułożenia, to znaleźliśmy numer właściwej ćwiartki.

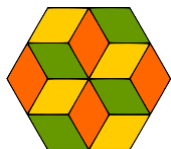
Pozostaje nam zastanowić się nad tym, jakie powinny być początkowe wartości zmiennych oraz jak w prosty sposób cyklicznie zmieniać numery ćwiartek.

Drugą kwestię rozwiążemy wprowadzając następującą numerację ćwiartek:

0 – I
1 – IV
2 – III
3 – II

Numeracja 0-3 odzwierciedla kolejność przechodzenia ćwiartek. Natomiast przejście z ostatniej ćwiartki (oznaczonej 3) do pierwszej ćwiartki (oznaczonej 0) zrealizujemy operacją modulo:

```
cwiartka = (cwiartka + 1) % 4
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Pełny zapis algorytmu wraz z inicjalizacją zmiennych może wyglądać tak:

```
numer = 0
cwiartka = 3
ile_d = 0
dopóki numer < n
    cwiartka = (cwiartka + 1) % 4
    jeśli cwiartka == "I" lub cwiartka == "II"
        ile_d ← ile_d + 1
    numer = numer + ile_d
    jeśli numer >= n
        wynik ktora(cwiartka)
```

Funkcja **ktora** z parametrem **cwiartka** odtwarza pierwotne oznaczenie ćwiartek.

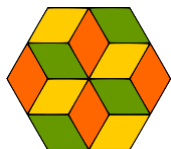
Rozwiązanie w języku Python

Funkcja realizująca opisany algorytm, zapisana w języku Python może wyglądać tak:

```
1. def klocki(n):
2.     numer = 0
3.     cwiartka = 3
4.     ile = 0
5.     while numer < n:
6.         cwiartka = (cwiartka + 1) % 4
7.         if cwiartka == 0 or cwiartka == 3:
8.             ile = ile + 1
9.         numer = numer + ile
10.        if numer >= n:
11.            return ktora(cwiartka)
```

Zamianę numeru ćwiartki z notacji 0-3 (zgodnej z ruchem wskazówek zegara) na oznaczenie ćwiartek z treści zadania (notacja matematyczna) realizuje poniższa funkcja:

```
12. def ktora(cwiartka):
13.     if cwiartka == 0: return 'I'
14.     if cwiartka == 1: return 'IV'
15.     if cwiartka == 2: return 'III'
16.     if cwiartka == 3: return 'II'
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie alternatywne

Istnieje rozwiązanie wyznaczające odpowiedź bez użycia iteracji. W rozwiązaniu tym wyznaczany jest numer warstwy, w której leży rozpatrywany klocek:

numery klocków	numer warstwy	
1 – 4	0	
5 – 16	1	
17 – 36	2	
37 – 64	3	

Numer warstwy dla klocka o numerze n obliczymy ze wzoru:

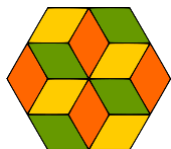
$\text{int}(\text{sqrt}((n - 1) // 4))$

Na podstawie przynależności klocka do warstwy ustalana jest jego pozycja (położenie w danej ćwiartce). Należy przy tym zauważyć, że:

- w warstwach poniżej rozpatrywanej leży $4 * w * w$ klocków;
- w każdej ćwiartce w danej warstwie leży $2 * w + 1$ klocków.

Rozwiązanie w języku Python

```
1. from math import sqrt
2.
3. def kloki(n):
4.     w = int(sqrt((n - 1) // 4)) #warstwa
5.     p = n - 4 * w * w - w       #pozycja
6.     if p <= 0:
7.         return 'II'
8.     p = p - (2 * w + 1)
9.     if p <= 0:
10.        return 'I'
11.    p = p - (2 * w + 1)
12.    if p <= 0:
13.        return 'IV'
14.    p = p - (2 * w + 1)
15.    if p <= 0:
16.        return 'III'
17.    return 'II'
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Pierwsza grupa testów sprawdza poprawność dla wartości z rysunku pomocniczego. Wartości te leżą na przekątnych ćwiartek. Druga i trzecia grupa testów została tak dobrana, by sprawdzić poprawność przekraczania ćwiartek (druga grupa dla liczb niewiele przekraczających wartości z rysunku pomocniczego, trzecia grupa dla liczb znacznie je przekraczających). Czwarta i piąta grupa testów miała na celu wyeliminowanie rozwiązań typu jeżeli ... to ...

Grupa testów	Test	Wynik
I	klocki(7)	I
	klocki(10)	IV
	klocki(13)	III
	klocki(16)	II
II	klocki(46)	I
	klocki(47)	IV
III	klocki(333)	II
	klocki(334)	I
IV	klocki(9019)	IV
	klocki(9230)	II
V	klocki(9891)	III
	klocki(9999)	II