Programowanie w języku Python – ćwiczenia 9

Zagadnienia: własności liczb całkowitych

Liczby doskonałe to takie liczby naturalne, które są sumą wszystkich swoich dzielników właściwych (tj. liczb mniejszych od dzielnej). Są nimi np. `6 = 1 + 2 + 3`, oraz
`28 = 1 + 2 + 4 + 7 + 14`.
Liczby zaprzyjaźnione to para **różnych** liczb całkowitych dodatnich takich, że suma dzielników właściwych każdej z tych liczb równa się drugiej liczbie. Pierwszą parę takich liczb podał Pitagoras – to `220` i `284`.

`220 = 1 + 2 + 4 + 71 + 142` (dzielniki `284`)
`284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110` (dzielniki `220`)

Zadanie 1

Zdefiniuj jednoparametrową funkcję logiczną `czy_doskonala(liczba)`, zwracającą `True` jeśli `liczba` jest doskonała, oraz `False` w przeciwnym przypadku.

Zadanie 2

Zdefiniuj jednoparametrową funkcję `wypisz_doskonale(n)`, gdzie `n` oznacza zakres, w którym będą wypisywane liczby doskonałe. Skorzystaj z napisanej wcześniej funkcji `czy_doskonala(liczba)`.

Zadanie 3

Napisz program, który wyznaczy wszystkie trzycyfrowe liczby doskonałe.

Zadanie 4

Zdefiniuj funkcję logiczną `zaprzy(a,b)`, która w wyniku zwróci `True`, jeśli para liczba `a` i `b` podanych jako parametry będzie zaprzyjaźniona, lub `False` – jeśli nie będzie.

Zadanie 5

Zdefiniuj jednoparametrową funkcję `wypisz_zaprzy(n)`, gdzie `n` oznacza zakres, w którym będą wypisywane liczby zaprzyjaźnione. Dla `n=10000` istnieje pięć par liczb zaprzyjaźnionych: `220` i `284`, `1184` i `1210`, `2620` i `2924`, `5020` i `5564` oraz `6232` i `6368`.

Zadanie 6

Liczba jest k-doskonała, jeśli różni się od sumy wszystkich swoich dzielników właściwych o k. Dzielniki właściwe liczby to dzielniki mniejsze od tej liczby. Na przykład, liczba `10` jest `2`-doksonała, bo suma jej dzielników właściwych `1 + 2 + 5 = 8`, różni się od niej o `2`. Napisz funkcję logiczną `k_doskonala(n,k)`, która sprawdzi, czy podana liczba naturalna `n` jest liczbą k-doskonałą. Funkcja zwróci odpowiednio `True` lub `False`.

Zadanie 7

Zdefiniuj funkcję logiczną `czy_pierwsza(n)`, której parametrem jest liczba naturalna `n`, a wynikiem wartość `True`, gdy jest ona liczbą pierwszą, albo `False`, gdy nią nie jest.

Zadanie 8

Zdefiniuj funkcję `pierwsza(n)`, której parametrem będzie liczba naturalna `n`, a wynikiem – `n`-ta liczba pierwsza.

Zadanie 9
Napisz program, który wyznaczy wszystkie trzycyfrowe liczby pierwsze.

Zadanie 10
Napisz funkcję, która zwróci największą liczbę pierwszą mniejszą od dodatniej liczby całkowitej podanej jako argument funkcji.

Zadanie 11
Napisz funkcję, która zwróci w tablicy wszystkie dzielniki pierwsze liczby podanej jako argument funkcji.

Zadanie 12
Dwie liczby pierwsze różniące się o `2` nazywamy bliźniaczymi. Zdefiniuj funkcję `blizniacze(n)`, której parametrem jest liczba naturalna `n`, a wynikiem – pierwsza liczba z `n`-tej pary liczb bliźniaczych.

Zadanie 13
Liczbę naturalną nazywamy super-pierwszą, jeśli jest liczbą pierwszą oraz suma jej cyfr (w systemie dziesiętnym) jest też liczbą pierwszą. Taką liczbą jest np. `101`. Napisz funkcję logiczną `super_piewsza(n)`, która sprawdzi, czy podana liczba naturalna `n` jest liczbą super-pierwszą. Funkcja zwróci odpowiednio `True` lub `False`.

Zadanie 14
Liczby czworacze to cztery liczby pierwsze postaci `n, n+2, n+6, n+8`, np. `5, 7, 11, 13` lub `11, 13, 17, 19` lub `101, 103, 107, 109`. Zdefiniuj funkcję `czworacze(n)`, której parametrem będzie liczba naturalna `n`, a wynikiem pierwsza liczba z liczb czworaczych, większa od podanego parametru.

Zadanie 15
Napisz program wyświetlający liczby, których pierwiastki mieszczą się w podanym przez użytkownika przedziale. Na przykład dla przedziału `[5;6]` są to liczby: `25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36`.

Zadanie 16
Liczba `362881` ma tę własność, że przy dzieleniu przez `2,3,4,5,6,7,8,9` daje resztę `1`. Znajdź najmniejszą liczbę o tej własności.

Zadanie 17
Ciąg liczb Collatza zdefiniowany jest następująco: pierwsza liczba ciągu jest dowolną liczbą naturalną `x`, każda kolejna wartość ciągu obliczana jest na podstawie poprzedniej według poniższej zasady

- jeśli poprzednia wartość była parzysta, to należy podzielić ją przez `2`,
- jeśli poprzednia wartość była nieparzysta, to należy pomnożyć ją przez `3` i dodać `1`.

Wobec tego dla wartości początkowej `x=10` kolejne liczby to: `5,16, 8, 4, 2, 1`. Zdefiniuj funkcję `collatz(x)`, której parametrem będzie liczba naturalna `x`, czyli wartość początkowa ciągu liczb Collatza, a wynikiem – liczba kroków, po których w ciągu pojawi się liczba `1`.