

Завдання 3 А

9. Знайти k перших чисел Фібоначчі, що більші заданого числа A і вивести кількість різних цифр, які зустрічаються у кожному числі. Тут можна використати список.

Для виконання даного завдання використовуємо оператори циклу `while`, `for`, умовні оператори `if`, `else`.

Алгоритм

- 1) Користувач вводить число, з якого він бажає розпочати пошук чисел послідовності Фібоначчі та їх кількість. Робимо початкову ініціалізацію кількості знайдених чисел « $n=0$ », першого та другого чисел Фібоначчі, які обоє дорівнюють 1.
- 2) Розпочинаємо цикл `while` з умовою, що кількість знайдених чисел не перевищує кількість потрібних. Наступне число F_i задаємо як суму F_1 і F_2 . Так буде утворюватись послідовність Фібоначчі.
- 3) За допомогою умовного оператора `if`, створюємо умову, яка перевіряє чи F_i більше, ніж число з якого користувач бажав почати пошук. Якщо умова виконується, то збільшуємо кількість знайдених чисел n на 1. Перетворюємо число на стрічку. Константі m присвоюємо значення довжини стрічки.
- 4) Створюємо новий список `data`, який складається з десяти нулів, кожен з яких символізує цифру(0-9). За допомогою оператора циклу `for` присвоюємо змінній `digit` значення однієї з цифр числа. Якщо така цифра трапляється в числі один раз, то в масиві `data` елемент з індексом, який дорівнює цій цифрі, стає 1. З кожним повторенням цифри, значення елемента в масиві збільшується на 1.
- 5) Створюємо новий список `new`, в який додаємо всі елементи з масиву `data`, які більші за 0. Це буде діапазон різних цифр в числі, а його довжина- кількістю різних цифр в числі.
- 6) Генеруємо наступне число Фібоначчі.

Лістинг

```
A=int(input("Введіть число, з якого розпочнеться пошук:"))
k=int(input("Введіть кількість потрібних чисел:"))
F1=1
F2=1
n=0
while n<k:
    Fi=F1+F2
    if Fi>A:
        n=n+1
        mystr=list(str(Fi))
        m=len(mystr)
        data=[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        # [redacted]
        # [redacted]
        # [redacted]
        # [redacted]
        # [redacted]
        new=[]
        for i in range(0,10):
            if data[i]!=0:
                new.append(i)
        print(Fi, '-', len(new), 'цифр', new)
    F1=F2
    F2=Fi
```

Приклади виконання

1. Введіть число, з якого розпочнеться пошук:55

Введіть кількість потрібних чисел:7

89 - 2 цифр [8, 9]

144 - 2 цифр [1, 4]

233 - 2 цифр [2, 3]

377 - 2 цифр [3, 7]

610 - 3 цифр [0, 1, 6]

987 - 3 цифр [7, 8, 9]

1597 - 4 цифр [1, 5, 7, 9]

2. Введіть число, з якого розпочнеться пошук:144

Введіть кількість потрібних чисел:14

233 - 2 цифр [2, 3]

377 - 2 цифр [3, 7]

610 - 3 цифр [0, 1, 6]

987 - 3 цифр [7, 8, 9]

1597 - 4 цифр [1, 5, 7, 9]

2584 - 4 цифр [2, 4, 5, 8]

4181 - 3 цифр [1, 4, 8]

6765 - 3 цифр [5, 6, 7]

10946 - 5 цифр [0, 1, 4, 6, 9]

17711 - 2 цифр [1, 7]

28657 - 5 цифр [2, 5, 6, 7, 8]

46368 - 4 цифр [3, 4, 6, 8]

75025 - 4 цифр [0, 2, 5, 7]

121393 - 4 цифр [1, 2, 3, 9]

Завдання 3 В

9. Підрахувати кількість натуральних простих чисел, що належать інтервалу $[a,b]$, в яких кількість парних і непарних цифр однакова. 0-парне. Вивести всі прості числа з даного інтервалу $[a,b]$ з позначкою true/false у випадку співпадіння/не співпадіння кількості цифр.

Для виконання даного завдання використовуємо оператори циклу while, for, умовні оператори if ,else.

Алгоритм

- 1) Користувач вводить інтервал, в якому хоче шукати прості числа. Кількість знайдених чисел, в яких кількість парних і непарних цифр однакова позначаємо $k=0$. Числу number, з якого починаємо пошук простих чисел, присвоюємо значення

- a. Якщо a менше 3, то одразу виводимо два перші прості числа- 1 і 2.
- 2) За допомогою while створюємо цикл, умовою якого є те, що число number менше, ніж b. Створюємо змінну counter, якій спочатку присвоюємо значення 0.
- 3) За допомогою циклу for, умовного оператора if та змінної counter перевіряємо чи число просте.
- 4) За допомогою умовного оператора if та оператора циклу while рахуємо кількість парних і непарних цифр в числі, ділячи число на 2 для визначення парності останньої цифри та на 10 для зменшення числа на одну цифру. Виводимо число на екран з позначками True /False відповідно до того, чи кількість парних і непарних цифр в числі однакова.
- 5) Генеруємо наступне число.

Лістинг

```
import math

a=int(input("Enter a= "))

b=int(input("Enter b= "))

k=0

number=a

if a<3:

    number=3

    print("1 False", "2 False", sep="\n")

while number<=b:

    counter = 0

    for i in range (2,math.ceil((math.sqrt(number)))+1):

        if number%i!=0:

            continue
```


1. Enter a= 1

Enter b= 17

1 False

2 False

3 False

5 False

7 False

11 False

13 False

17 False

Кількість чисел з однаковою кількістю парних і непарних цифр 0

2. Enter a= 4789

Enter b= 4960

4789 True

4793 False

4799 False

4801 False

4813 True

4817 True

4831 True

4861 False

4871 True

4877 True

4889 False

4903 True

4909 True

4919 False

4931 False

4933 False

4937 False

4943 True

4951 False

4957 False

Кількість чисел з однаковою кількістю парних і непарних цифр 9