

# Функции

## Цели практической работы

Научиться:

- разбивать программу на функции и выделять блоки, которые могут быть использованы повторно;
- передавать значения, объекты в функции через аргументы;
- использовать двойное сравнение (chained comparison).

### Задача 1. Сумма чисел

Напишите функцию `summa_n`, которая принимает одно целое положительное число `N` и выводит сумму всех чисел от 1 до `N` включительно.

Пример работы программы:

Введите число: 5

Я знаю, что сумма чисел от 1 до 5 равна 15

```
number = int(input('Введите число: '))
def summa_n(number):
    summ = 0
    for i_num in range(1, number + 1):
        summ += i_num
    print('Я знаю, что сумма чисел от 1 до', number, 'равна ', summ)
summa_n(number)
```

### Задача 2. Функция в функции

Евгений проходит специальный тест по программированию. Всё шло хорошо, пока герой не наткнулся на тему «Функции». Задание звучит так:

Основная ветка программы, не считая заголовков функций, состоит из одной строки кода. Это вызов функции `test()`. В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция `positive()`, тело которой содержит команду вывода на экран слова «Положительное». Если число отрицательное, то вызывается функция `negative()`, её тело содержит выражение вывода на экран слова «Отрицательное».

Помогите Евгению и реализуйте такую программу.

```
def positive():
    print('Положительное')
def negative():
    print('Отрицательное')

number = int(input('Введите целое число: '))
def test(number):
    if number > 0:
```

```

    positive()
elif number < 0:
    negative()
else:
    print('Ошибка ввода.')
test(number)

```

### Задача 3. Апгрейд калькулятора

Степан использует калькулятор для расчёта суммы и разности чисел, но на работе ему требуются не только обычные арифметические действия. Он ничего не хочет делать вручную, поэтому решил немного расширить функционал калькулятора.

Напишите программу, запрашивающую у пользователя число и действие, которое нужно сделать с числом: вывести сумму его цифр, максимальную или минимальную цифру. Каждое действие оформите в виде отдельной функции, а основную программу зациклите.

Запрошенные числа должны передаваться в функции суммы, максимума и минимума при помощи аргументов.

```

def summ_number(number):
    summ = 0
    while number > 0:
        summ += number % 10
        number //= 10
    print('Сумма цифр: ', summ)

def max_number(number):
    max_num = 0
    while number > 0:
        if number % 10 > max_num:
            max_num = number % 10
        number //= 10
    print('Максимальная цифра: ', max_num)

def min_number(number):
    min_num = 9
    while number > 0:
        if number % 10 < min_num:
            min_num = number % 10
        number //= 10
    print('Минимальная цифра: ', min_num)

while True:
    number = int(input('Введите число: '))
    action = int(input('Введите номер действия: 1 - сумма цифр, 2 -
максимальная цифра, 3 - минимальная цифра \n'))
    if action == 1:
        summ_number(number)
    elif action == 2:
        max_number(number)
    elif action == 3:

```

```
min_number(number)
else:
    print('Ошибка ввода')
print()
```

#### Задача 4. Число наоборот

Вводится последовательность чисел, оканчивающаяся нулём. Реализуйте функцию, которая принимает в качестве аргумента каждое число, переворачивает его и выводит на экран.

Пример:

Введите число: 1234

Число наоборот: 4321

Введите число: 1000

Число наоборот: 0001

Введите число: 0

Программа завершена!

```
def new_number():
    num = input("Введите целое число: ")
    new_number = num[::-1]
    if num == "0":
        print("Программа завершена!")
    else:
        print(f"Число наоборот: {new_number}", end='')

new_number()
```

*Дополнительно: добейтесь такого вывода чисел, в начале которых идут нули.*

Пример:

Введите число: 1230

Число наоборот: 321

Ноль, который мы убрали, называется ведущим.

```
def new_number():
    num = int(input("Введите целое число: "))
    new_num = 0

    while num != 0:
        for i in range(num):
            last = num % 10
            if last == 0:
                num = num // 10
            else:
                num = num // 10 # отбрасываем ее
                new_num = new_num * 10 + last # добавляем следующую цифру
        print(f"Число наоборот: {new_num}", end='\n')
```

```

new_number()

else:
    print("Программа завершена!")

new_number()

```

## Задача 5. Текстовый редактор

Продолжаем разрабатывать новый текстовый редактор. В этот раз нам поручили написать для него код, который считает, сколько раз в тексте встречается любая выбранная буква или цифра (а не только буквы Ы, как раньше).

Напишите функцию `count_letters()`, которая принимает на вход текст и подсчитывает, какое в нём количество цифр К и букв N. Функция должна вывести на экран информацию о найденных буквах и цифрах в определённом формате.

Пример:

Введите текст: 100 лет в обед

Какую цифру ищем? 0

Какую букву ищем? Л

Количество цифр 0: 2

Количество букв Л: 1

```

text = input('Введите последовательность чисел: ')

def count_letters(text, number, letter):
    count_number = 0
    count_letter = 0

    for i in text:
        if i == number:
            count_number += 1
        elif i == letter:
            count_letter += 1
    print('Количество цифр ', number, ':', count_number)
    print('Количество букв ', letter, ':', count_letter)

number = input('Какую цифру ищем? ')
letter = input('Какую букву ищем? ')
count_letters(text, number, letter)

```

## Задача 6. НОД

Напишите функцию, вычисляющую наибольший общий делитель двух чисел.

```

def gcd (a, b):
    while a != 0 and b != 0:
        if a > b:
            a = a % b
        else:
            b = b % a

```

```
print ('Наибольший общий делитель:', a + b)

gcd(30,18)
```

## Задача 7. Неделка

Вы пришли на работу в компанию по разработке игр, целевая аудитория — дети и их родители. У предыдущего программиста было задание сделать две игры в одном приложении, чтобы пользователь мог выбирать одну из них. Однако программист, на место которого вы пришли, перед увольнением не успел выполнить эту задачу и оставил только небольшой шаблон проекта. Используя этот шаблон, реализуйте игры «Камень, ножницы, бумага» и «Угадай число».

Правила игры «Камень, ножницы, бумага»: программа запрашивает у пользователя строку и выводит, победил он или проиграл. Камень бьёт ножницы, ножницы режут бумагу, бумага кроет камень.

Правила игры «Угадай число»: программа запрашивает у пользователя число до тех пор, пока он не отгадает загаданное.

```
def rock_paper_scissors():
    # Здесь будет игра «Камень, ножницы, бумага»

def guess_the_number():
    # Здесь будет игра «Угадай число»

def mainMenu():
    # Здесь главное меню игры

mainMenu():
    pass
```

```
def rock_paper_scissors():
    print('Игра "Камень, ножницы, бумага"')
    print('Введите "камень", "ножницы" или "бумага"')
    user_choice = input('Ваш выбор: ')
    if user_choice == 'камень':
        print('Ничья')
    elif user_choice == 'ножницы':
        print('Вы проиграли')
    elif user_choice == 'бумага':
        print('Вы выиграли')
    else:
        print('Ошибка ввода')
        print()
    main_menu()

def guess_the_number():
    print('Угадай число!')
    while True:
        number = int(input('Введите число: '))
        if number < 5:
            print('Число меньше, чем нужно. Попробуйте ещё раз!')
```

```
elif number > 5:
    print('Число больше, чем нужно. Попробуйте ещё раз!')
else:
    print('Вы угадали!')
    print()
    main_menu()

def main_menu():
    print('Выберете игру: 1 - Камень, ножницы, бумага; 2 - Угадай число')
    choice = int(input('Введите номер игры: '))
    if choice == 1:
        rock_paper_scissors()
    elif choice == 2:
        guess_the_number()
    else:
        print('Ошибка ввода: нужно ввести 1 или 2')
        print()

main_menu()
```