

## Процедуры

```
# -*- coding: utf-8 -*-  
"""
```

Программа к учебнику информатики для 10 класса  
К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 20. Программа с процедурой

Вход:

-1

Результат:

Ошибка программы

```
"""
```

```
def Error():
```

```
    print ( "Ошибка программы" )
```

```
n = int( input() )
```

```
if n < 0:
```

```
    Error()
```

```
# -*- coding: utf-8 -*-  
"""
```

Программа к учебнику информатики для 10 класса  
К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 21. Процедура с параметром

Вход:

99

Результат:

Двоичный код: 01100011

```
"""
```

```
def printBin(n):
```

```
    k = 128
```

```
    while k > 0:
```

```
        print(n // k, end="")
```

```
        n = n % k
```

```
        k = k // 2
```

```
n = int ( input("Введите натуральное число: ") )
```

```
print ( "Двоичный код: ", end="")
```

```
printBin ( n )
```

```
# -*- coding: utf-8 -*-
```

""""

Программа к учебнику информатики для 10 класса

К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 21. Процедура с параметром

Вход:

99

Результат:

Двоичный код: 01100011

""""

```
def printBin(n):
    k = 128
    while k > 0:
        print(n // k, end="")
        n = n % k
        k = k // 2

n = int ( input("Введите натуральное число: ") )
print ( "Двоичный код: ", end="" )
printBin ( n )
```

**A**

«**A**»: Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран линию из N символов '–'.

**Пример:**

**Введите N:**

**10**

-----

```
# -*- coding: utf-8 -*-
```

```
def Error():
    for i in range(n):
        print('-', end="")
```

```
n = int( input() )
```

```
if n > 0:
```

```
    Error()
```

**B**

«**B**»: Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с первой.

**Пример:**

**Введите натуральное число:**

**1234**

**1**

2  
3  
4

```
# -*- coding: utf-8 -*-  
def Er():  
    for i in n:  
        print(i)  
  
n = (input())  
Er()
```

C

«C»: Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.

**Пример:**

**Введите натуральное число:**

**2013**

**MMXIII**

```
def Arabic2Roman(number):  
    arab = [1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000];  
    roman = ['I', 'IV', 'V', 'IX', 'X', 'XL', 'L', 'XC', 'C', 'CD', 'D', 'CM', 'M'];  
    if number == 0:  
        return ""  
    result = ""  
    pos = len(arab) - 1;  
    while number > 0:  
        if number >= arab[pos]:  
            result += roman[pos]  
            number -= arab[pos]  
        else:  
            pos -= 1  
    return result  
n = int(input('Введите значение: '))  
print(Arabic2Roman(n))
```

**Функции**

```
# -*- coding: utf-8 -*-  
"""
```

Программа к учебнику информатики для 10 класса

К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 23. Функция

Вход:

12345

Результат:

Сумма цифр 15

"""

```
def sumDigits(n):
```

```
    sum = 0
```

```
    while n != 0:
```

```
        sum += n % 10
```

```
        n = n // 10
```

```
    return sum
```

```
N = int ( input("Введите целое число: ") )
```

```
print ( "Сумма цифр", sumDigits(N))
```

```
# -*- coding: utf-8 -*-
```

"""

Программа к учебнику информатики для 10 класса

К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 24. Логическая функция

Вход:

15

Результат:

Простые числа: 2 3 5 7 11 13

"""

```
def isPrime(n):
```

```
    count = 0
```

```
    k = 2
```

```
    while k*k <= n and n % k != 0:
```

```
        k += 1
```

```
    return k*k > n
```

```
M = int( input("Введите максимальное число: ") )
```

```
print ( "Простые числа: ", end="" )
```

```
for i in range(2, M+1):
```

```
    if isPrime(i):
```

```
        print ( i, " ", end="", sep="" )
```

```
# -*- coding: utf-8 -*-
```

"""

Программа к учебнику информатики для 10 класса

К.Ю. Полякова и Е.А. Еремина.

Глава 8.

Программа № 25. Логическая функция

Вход:

```

5
7
12
Результат:
5 - простое число
7 - простое число
""""
def isPrime(n):
    count = 0
    k = 2
    while k*k <= n and n % k != 0:
        k += 1
    return k*k > n

n = int( input("Введите число: ") )
while isPrime ( n ):
    print ( n, "- простое число" )
    n = int( input("Введите число: ") )

```

**А**  
**«А»:** Напишите функцию, которая находит наибольший общий делитель двух натуральных чисел.

**Пример:**  
**Введите два натуральных числа:**  
**7006652 112307574**  
**НОД(7006652,112307574) = 1234.**

```

def gcd(a : int, b : int):

    while(a != 0 and b != 0):

        if (a > b): a = a % b

        else: b = b % a

    return a + b

print(gcd(7006652, 112307574))

```

**В**  
**«В»:** Напишите функцию, которая определяет сумму цифр переданного ей числа.  
**Пример:**  
**Введите натуральное число:**  
**123**  
**Сумма цифр числа 123 равна 6.**

```
def s(a):
    result = 0
    while a > 0:
        result += a % 10
        a //= 10
    return result
n = int( input() )
print(s(n))
```

**С**

«С»: Напишите функцию, которая «переворачивает» число, то есть возвращает число, в котором цифры стоят в обратном порядке.

**Пример:**

**Введите натуральное число:**

**1234**

**После переворота: 4321.**

```
def rev(n):
    r = 0
    while(n>0):
        r = r*10+n%10
        n = n//10
    return r
n=int(input())
r=rev(n)
print(r)
```

**А**

«А»: Напишите функцию, которая переставляет три переданные ей числа в порядке возрастания.

**Пример:**

**Введите три натуральных числа:**

**10 15 5**

**5 10 15**

```
arg = list(map(int, input().split()))
def srt(arg):
    return print(sorted(arg))
srt(arg)
```

**В**

«В»: Напишите функцию, которая сокращает дробь вида М/Н.

**Пример:**

**Введите числитель и знаменатель дроби:**

**25 15**

**После сокращения: 5/3**

```

def ReduceFraction(n, m):
    if n > m:
        k = n
    else:
        k = m
    while k != 1:
        if n % k == 0 and m % k == 0:
            return n // k, m // k
        else:
            k -= 1
    return n, m
n, m = int(input()), int(input())
print(ReduceFraction(n, m))

```

**С**

«С»: Напишите функцию, которая вычисляет наибольший общий делитель и наименьшее общее кратное двух натуральных чисел.

**Пример:**

**Введите два натуральных числа:**

**10 15**

**НОД(10,15)=5**

**НОК(10,15)=30**

```

import math
data = list(map(int, input().split()))
gcd = math.gcd(*data)
lcm = math.lcm(*data)
print ("НОД:",gcd,"НОК:", lcm)

```

```

import math
data = list(map(int, input().split()))
def srt(data):
    gcd = math.gcd(*data)
    lcm = math.lcm(*data)
    return print ("НОД:",gcd,"НОК:", lcm)
srt(data)

```

**А**

«А»: Напишите логическую функцию, которая определяет, является ли переданное ей число совершенным, то есть, равно ли оно сумме своих делителей, меньших его самого.

**Пример:**

**Введите натуральное число:**

**28**

Число 28 совершенное.

Пример:

Введите натуральное число:

29

Число 29 не совершенное.

```
# Уровень А

def is_perfect_number(number):
    for i in range(2, number):
        if number % i == 0:
            return 'Число '+str(number)+' совершенное'
            break
        else:
            continue
    return 'Число '+str(number)+' несовершенное'
```

**В**

«В»: Напишите логическую функцию, которая определяет, являются ли два переданные ей числа взаимно простыми, то есть, не имеющими общих делителей, кроме 1.

Пример:

Введите два натуральных числа:

28 15

Числа 28 и 15 взаимно простые.

Пример:

Введите два натуральных числа:

28 16

Числа 28 и 16 не взаимно простые.

```
# Уровень В

def is_prime_number(a, b):
    n1 = str(a)
    n2 = str(b)
    while a != 0 and b != 0:
        if a > b:
            a = a % b
        else:
            b = b % a

    if a + b == 1:
        return 'Числа '+n1+' и '+n2+' взаимно простые'
    else:
        return 'Числа '+n1+' и '+n2+' не взаимно простые'
```

**С**

**«С»:** Простое число называется гиперпростым, если любое число, получающееся из него откидыванием нескольких цифр, тоже является простым. Например, число 733 – гиперпростое, так как и оно само, и числа 73 и 7 – простые. Напишите логическую функцию, которая определяет, верно ли, что переданное ей число – гиперпростое. Используйте уже готовую функцию isPrime, которая приведена в учебнике.

**Пример:**

**Введите натуральное число:**

**733**

**Число 733 гиперпростое.**

**Пример:**

**Введите натуральное число:**

**19**

**Число 19 не гиперпростое.**

```
def isPrime(X):
    x = str(X)
    n = 2
    i = 0
    d = 10
    while X >= n:
        i += 1
        n *= 10
    if i == 3:
        num = int(x[i-1])
        if num % 2 != 0:
            D = X // 10
            if D % 2 != 0:
                print("Да, число гиперпростое.")
        else:
            print("Нет, число не гиперпростое.")

isPrime(737)
```