Реализация задач на языке программирования Python

При написании программ часто возникает ситуация, когда необходимо производить различные математические вычисления. Как и другие языки программирования, Python предоставляет разнообразные функции для выполнения вычислений.

Знакомство с библиотеками

1. Теоретический материал

Для работы с математическими функциями нужно импортировать библиотеку **math:**

```
import math
```

После этого к функциям из этой библиотеки можно обращаться следующим образом:

```
math.имя_функции(…)
```

Далее в таблице приведены синтаксис и описание основных математических функций языка Python

ceil(x)	Возвращает округленное *x* как ближайшее целое значение типа int**,** большее или равное *x* (округление "вверх").
floor(x)	В противоположность ceil(x) возвращает округленное *x* как ближайшее целое значение типа int, меньшее или равное *x* (округление "вниз").
fabs(x)	Возвращает абсолютное значение (модуль) числа *x*.
exp(x)	Возвращает e^x.
log(x[, base])	При передаче функции одного аргумента *x*, возвращает натуральный логарифм x (логарифм по основанию e = 2.718…). При передаче двух аргументов, второй берется как основание логарифма.
pow(x, y)	Возвращает *x* в степени *y*.
sqrt(x)	Квадратный корень из *x*.
acos(x)	Возвращает арккосинус *x*, в радианах.
log(x[, base])	При передаче функции одного аргумента *x*, возвращает натуральный логарифм *x* (логарифм по основанию *e* = 2.7182…). При передаче двух аргументов, второй берется как основание логарифма.
asin(x)	Возвращает арксинус *x*, в радианах.
atan(x)	Возвращает арктангенс *x*, в радианах.
cos(x)	Возвращает косинус *x*, где *x* выражен в радианах.
sin(x)	Возвращает синус *x*, где *x* выражен в радианах.
tan(x)	Возвращает тангенс *x*, где *x* выражен в радианах.

2. Пример

Задача:

Для введенных чисел *x* и *y* найти значение функции

$$f(x,y) = 2y^x + \ln|x+y^3|$$

Решение (код программы):

```
import math
x = float(input('Введите x '))
y = float(input('Введите y '))
f = 2 * math.pow(y, x) + math.log(math.fabs(x + y ** 3))
print('f = ', f)
```

Задача:

Для введенных чисел *x* и *y* найти значение функции

$$f(x,y) = \begin{cases} \sin(xe^y), xy \le -1 \\ \sqrt{|\cos(xy)|}, -1 < xy < 5 \\ x^2 + \mathrm{tg}y, xy \ge 5 \end{cases}$$

Решение (код программы):

```
import math
x = float(input('Введите x '))
y = float(input('Введите y '))
if x * y <= -1:
    f = math.sin(x * math.exp(y))
elif x * y >= 5:
    f = x * x + math.tan(y)
else:
    f = math.sqrt(math.fabs(math.cos(x * y)))
print('f = ', f)
```

Задача:

Вычислить значение функции $f(x) = \sin(x - e^2) + 3^x$ на отрезке $[x_n, x_k]$ с шагом *hx*

Решение (код программы):

```
import math
xn = float(input('Введите xn '))
xk = float(input('Введите xk '))
hx = float(input('Введите hx '))
x = xn #устанавливаем x в начало отрезка в xn
while x <= xk: #пока не дойдем до конца отрезка xk
    f = math.sin(x + math.exp(2)) + math.pow(3, x)
    print('x = ', x, ' f = ', f)
    x = x + hx #прибавляем к аргументу шаг
```

Задача*:

Вычислить значения функции

$$f(x,y)=\begin{cases}\sqrt[5]{y+x}, & \text{при } x+y\le 2;\\ |\sin x|^{y}, & \text{при } x+y>2.\end{cases}$$

При этом x изменяется в отрезке $0\le x\le 1$ с шагом $hx=0.2$; y изменяется в отрезке $1\le y\le 2$ с шагом $hy=0.5$.

Решение (код программы):

```
import math

ax, bx, hx = 0.0, 1.0, 0.2
ay, by, hy = 1.0, 2.0, 0.5

x = ax #устанавливаем x в начало отрезка в xn
while x <= bx: #пока не дойдем до xk
    y = ay #устанавливаем y в начало отрезка в yn
    while y <= by: #пока не дойдем до yk
        if x + y <= 2:
            f = math.pow(x + y, 1.0 / 5.0)
        else:
            f = math.pow(math.fabs(math.sin(x)), y)
        print('x: = ', x, 'y = ', y, 'f = ', f) # выводим результат
        # или print('x = {:.3}, y = {:.3}, f = {:.3}'.format(x,y,f))
        # или print(f'x = {x:.3}, y = {y:.3}, f = {f:.3}')
        y = y + hy #прибавляем к y шаг
    x = x + hx #прибавляем к x шаг
```

	3. Задания
1.	***Задача:***
	Для введенных чисел x и y найти значение функции $f(x,y) = \ln\left|\sin(x+y)\right|$
2.	***Задача:***
	Для введенных чисел x и y найти значение функции $$f(x,y) = \begin{cases} \text{arctg}\sqrt[3]{\lvert x-y\rvert}(xe^{y}), \sin(x+y) \le -0{,}5 \\ 3\log_3(\lvert xy\rvert), -0{,}5 < \sin(x+y) < 0{,}5 \\ x^3 + y^{1{,}5}, \sin(x+y) \ge 0{,}5 \end{cases}$$
3.	***Задача:***
	Вычислить значение функции $f(x) = \cos^3(e*x) + \sin\lvert x\rvert$ на отрезке [a, b] с шагом hx
4.	***Задача:***
	Вычислить значения функции $$f(x,y) = \begin{cases} \sqrt[3]{\sin(xe^{0.1y})}, \text{ при } x+y \le 2; \\ \left|\log_2(x+y)\right|, \text{ при } x+y > 2. \end{cases}$$ При этом x изменяется в отрезке $1 \le x \le 2.5$ с шагом $hx = 0.5$; y изменяется в отрезке $1 \le y \le 4$ с шагом $hy = 1$.

Представление чисел в Python

1. Теоретический материал

Язык Python поддерживает работу с двоичными разрядами (битами) целочисленных величин, где каждый бит числа рассматривается в отдельности. Для обеспечения этого в Python используются так называемые битовые или поразрядные операторы, которые реализуют общеизвестные битовые операции.

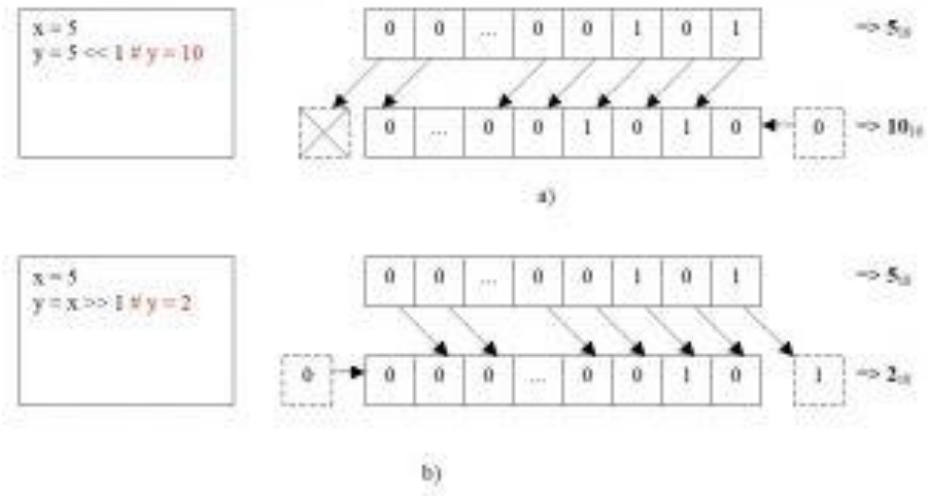
В битовых операторах (операциях) каждый операнд рассматривается как последовательность двоичных разрядов (бит), которые принимают значение 0 или 1 (двоичная система исчисления). Над этими разрядами можно выполнять известные операции (логическое «И», логическое «ИЛИ» и т.д.)

Перечень битовых операторов языка Python в порядке убывания приоритета следующий:

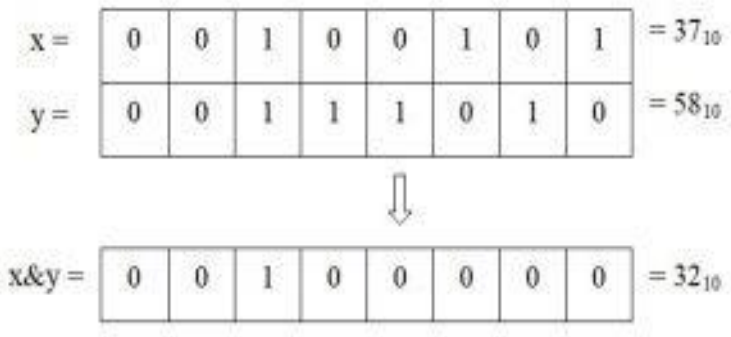
- ~ – битовый оператор НЕТ (инверсия, наивысший приоритет);
- <<, >> – операторы сдвига влево или сдвига вправо на заданное количество бит;
- **&** – битовый оператор И (AND);
- ^ – битовое исключающее ИЛИ (XOR);
- | – битовый оператор ИЛИ (OR).

В битовом операторе (операции) ~ инверсия значение любого бита числа изменяется на противоположное. Положительное число становится отрицательным со смещением -1, отрицательное число становится положительным со смещением на -1.

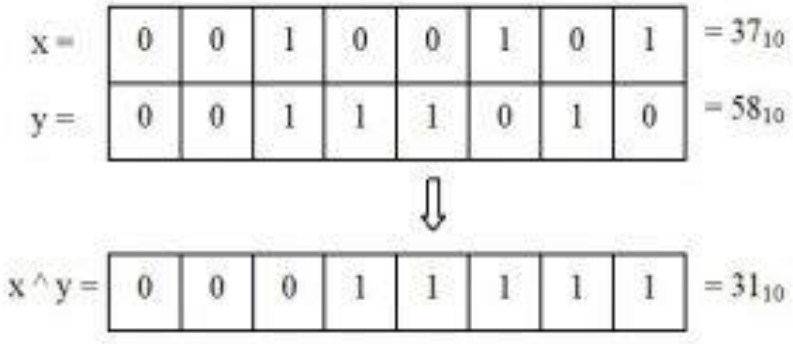
Операторы сдвига влево << и сдвига вправо >> сдвигают каждый бит на одну или несколько позиций влево или вправо.



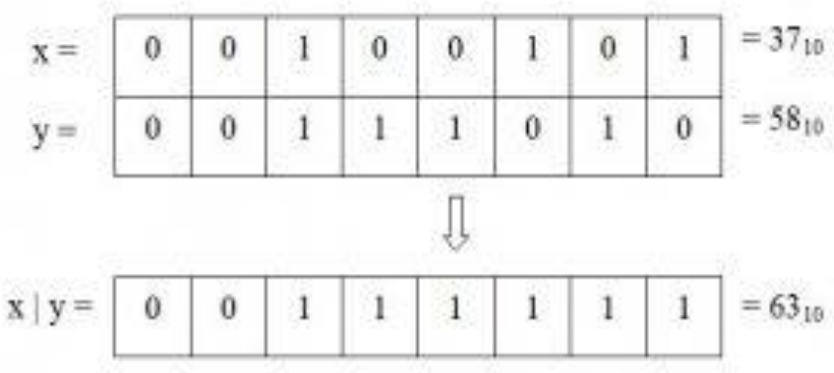
Битовый оператор И (AND) есть бинарным и выполняет побитовое «И» для каждой пары битов операндов, которые размещаются слева и справа от знака оператора &.



Битовый оператор исключительное ИЛИ обозначается символом ^ и выполняет операцию сложения по модулю 2 для любого бита операндов.



Битовый оператор ИЛИ (OR) символом |. Оператор реализует побитовое логическое сложение.



2. Пример

Задача:

Дано число **x = 37, y = 58**.
Найти **~x, x>>3, x<<2, x&y, x^y, x|y**

Решение (код программы):

```
x, y = 37, 58
#x в десятичной и двоичной системе
print('x = ', x, '   x_bin = ', bin(x))
#y в десятичной и двоичной системе
print('y = ', y, '   y_bin = ', bin(y))

#~x в десятичной и двоичной системе
a = ~x
print('~x =', a, '   ~x_bin = ', bin(a))

#x>>3 в десятичной и двоичной системе
b = x >> 3
print('x>>3 =', b, '   (x>>3)_bin = ', bin(b))

#x<<2 в десятичной и двоичной системе
c = x << 2
print('x<<2 =', c, '   (x<<2)_bin = ', bin(c))

#x&y в десятичной и двоичной системе
d = x & y
print('x&y =', d, '   (x&y)_bin = ', bin(d))

#x^y в десятичной и двоичной системе
e = x ^ y
print('x^y =', e, '   (x^y)_bin = ', bin(e))

#x|y в десятичной и двоичной системе
f = x | y
print('x|y =', f, '   (x|y)_bin = ', bin(f))
```

Задача:

Вытянуть из числа 4,5,6 биты и определить их целочисленное значение.

Решение (код программы):

```
number = int(input('Input number: '))
```

```
# фильтр на 4,5,6 биты
number &= 0b1110000
# сдвинуть на 4 разряда вправо
number >>= 4
print('number = ', number)
```

Задача:

Умножить значения двух чисел. В первом числе взять биты, которые размещенные в позициях 0-5. Во втором числе взять биты, которые размещены в позициях 0-7.

Решение (код программы):

```
x = int(input('x = '))
y = int(input('y = '))

# фильтр на 0-5 биты
x &= 0b11111

# фильтр на 0-7 биты
y &= 0b1111111

# умножить
z = x*y
print('x = ', x)
print('y = ', y)
print('z = ', z)
```

3. Задания
Задача:
Даны два различных числа k и n. Выведите значение $2^k + 2^n$, используя только битовые операции.
Задача:
Ввести число $n > 0$ с клавиатуры. Если число n является точной степенью двойки, вывести “YES”, в противном случае “NO”.
Задача:
Даны целые числа a и k. Выведите число, которое получается из a установкой значения k-го бита в 1.
Задача*:
Дано целые числа n и k. Обнулите в числе n его последние k бит и выведите результат. Рекомендуется сделать эту задачу без использования циклов.