

Рабочая тетрадь № 3

Для представления чисел в ЭВМ обычно используют битовые наборы – последовательности нулей и единиц фиксированной длины. Позиция в битовом наборе называется *разрядом*.

1. Теоретический материал

Для представления целых чисел без знака удобен битовый набор, соответствующий записи в двоичной системе счисления. Для целых чисел без знака как правило выделяют $k = 8, 16, 32$ или 64 разряда.

Для получения компьютерной записи целого числа без знака требуется его перевод в двоичную систему счисления, далее необходимо дополнить результат нулями слева до стандартной разрядности k .

2. Пример

Задача:

Найти представление беззнакового целого числа 26_{10} в восьмиразрядном битовом наборе

Решение:

Переведем число 26 в двоичную систему счисления.

$$\begin{array}{r} 26 \mid 2 \\ -26 \quad 13 \quad 2 \\ \hline 0 \quad -12 \quad 6 \quad 2 \\ \quad \quad 1 \quad -6 \quad 3 \quad 2 \\ \quad \quad \quad \quad 0 \quad -2 \quad 1 \\ \quad \quad \quad \quad \quad \quad 1 \end{array}$$

Результат перевода: $26_{10} = 11010_2$

Дополним полученный результат слева нулями до восьми шестнадцати
 $26_{10} = \mathbf{00011010}_2$

Ответ:

00011010

3. Задания

1. Задача:

Найти представление беззнакового числа 132_{10} в шестнадцатиразрядном битовом наборе

Решение:

132	2									
-132	66	2								
0	-66	33	2							
	0	-32	16	2						
		1	-16	8	2					
			0	-8	4	2				
				0	-4	2	2			
					0	-2	1			
						0				

00000000+10000100=0000000010000100

Ответ: 0000000010000100

2. Задача:

Найти минимальное и максимальное значения чисел для 16-ти разрядного беззнакового представления

Решение:

Ответ: 0000000000000000 = 0 – МИНИМАЛЬНОЕ
1111111111111111 = 65535- МАКСИМАЛЬНОЕ

1. Теоретический материал

Для целых чисел со знаком задействуют три варианта компьютерного представления:

- представление в прямом коде;
- представление в обратном коде;
- представление в дополнительном коде.

Во всех этих способах старший (левый) разряд равен нулю, если число положительное и единице, если число отрицательное. Остальные разряды числа (цифровая часть или мантисса) задействованы для представления модуля числа.

Положительные числа в дополнительном, обратном и прямом коде идентичны – мантисса включает двоичное представление числа, а в старшем разряде располагается ноль.

Для отображения отрицательного значения в прямом коде, в разряд знака ставится единица, а в разряды мантиссы – двоичный код его модуля.

Обратный код отрицательного числа получается инверсией всех цифр двоичного представления абсолютной величины, включая знаковый разряд: нули инвертируются в единицы, а единицы в нули.

Дополнительный код чисел с отрицательным знаком рассчитывается путем прибавления единицы к его младшему разряду обратного кода числа.

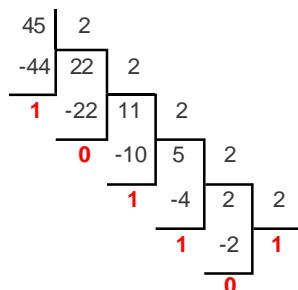
2. Пример

Задача:

Перевести число 45 в прямой, обратный и дополнительный код ($k = 8$)

Решение:

Сначала переведем десятичное число **45** в двоичную систему счисления.



Получилось: $45_{10} = 101101_2$.

Запишем **прямой код числа**. Первый слева разряд 0 (знак «плюс»). Оставшиеся 7 разрядов занимает число в двоичном представлении. Если в числе меньше 7 разрядов, оставшиеся дополняются нулями слева. Таким образом, для числа **45** получаем прямой код в виде **0,0101101** (первый слева 0 соответствует знаку, затем следует 0, дополняющий число до 7 разрядов, затем следует само двоичное число). Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково.

Ответ:

0,0101101

Задача:

Найти прямой, обратный и дополнительный коды в однобайтовом представлении для числа **-56**₁₀.

Решение:

Выполним перевод положительного числа **56** в двоичную систему счисления, получим: $56_{10} = 111000_2$.

Запишем **прямой код числа**. Всего в однобайтовом представлении 8 двоичных разрядов. Первый слева разряд — знаковый: 1 — для отрицательного числа. Оставшиеся 7 разрядов занимает число в двоичном представлении. Если в числе меньше 7 разрядов, оставшиеся дополняются нулями слева. Таким образом, для числа **-56** получаем прямой код в виде **1,0111000**.

Обратный код отрицательного числа получается из прямого инверсией всех разрядов, за исключением знакового. Получаем: **1,1000111**.

Дополнительный код отрицательного числа получается из обратного кода прибавлением к двоичному числу единицы (знаковый разряд в операции не участвует):

1000111

+ 1

1001000

Получаем: 1,1001000

Ответ:

Прямой код **1,0111000**

Обратный код **1,1000111**

Дополнительный код **1,1001000**

3. Задания

1. **Задача:**

Перевести число **12** в прямой, обратный и дополнительный код ($k = 8$)

Решение: $12_{10} = 1100_2$

Прямой код - $0,0001100_2$

Обратный код - $0,0001100_2$

Дополнительный код - $0,0001100_2$

Ответ

Прямой код - $0,0001100_2$

Обратный код - $0,1110011_2$

Дополнительный код - $0,1110100_2$

2. **Задача:**

Найти прямой, обратный и дополнительный коды в однобайтовом представлении для числа **-35**₁₀.

Решение: $35_{10} = 100011_2$

Прямой код - $1, 0100011$

Обратный код - $1,1011100$

Дополнительный код - $1,1011101$





Ответ:

Прямой код - $1, 0100011$

Обратный код - $1,1011100$

Дополнительный код - $1,1011101$

3. **Задача:**

	Задан дополнительный код числа в однобайтовом представлении: 1,1011100 . Найти число в десятичной системе счисления.
	Решение:
	$1,1011100_2 = -1 * 1011100$
	Доп код 1011100 = 1011011 (обратный) Обратный код 1011011 = 0100100 (прямой) $100100_2 = 4 + 32 = 36$ $-1 * 36 = -36_{10}$
	Ответ: -36
	

1. Теоретический материал

Сложение и вычитание чисел без знака осуществляется по стандартным алгоритмам для позиционных систем счисления.

При сложении в обратном коде складываются все разряды (включая знаковый) по обычному алгоритму. Результат сложения для k -разрядных чисел в общем виде имеет длину $k+1$ (старший разряд единица, если при сложении старших разрядов операндов был перенос, иначе – ноль). Значение левого $k+1$ -го разряда прибавляется к младшему разряду результата. В итоге получим k -разрядный битовый набор – сумму чисел в обратном коде. Разность чисел в обратном коде $x - y$ можно свести к операции сложения $x + (-y)$.

В дополнительном коде для сложения сначала по обычному алгоритму складываются все разряды (включая знаковый), а затем единицу переноса в $k+1$ -й разряд необходимо отбросить.

2. Пример

Задача:

Сложить два числа: $A_{10} = 7$, $B_{10} = 16$.

Решение:

Переведем числа в двоичную систему счисления

$$A_2 = +111 = +0111; B_2 = +10000.$$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$$[A_2]_{\text{п}} = [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0|00111; [B_2]_{\text{п}} = [B_2]_{\text{ок}} = [B_2]_{\text{дк}} = 0|10000.$$

Сложение в обратном или дополнительном коде дает один и тот же результат:

$$\begin{array}{r} 0| 00111 \\ + 0| 10000 \\ \hline C_2 = 0| 10111 \\ C_{10} = 23 \end{array}$$

Ответ:

23

Задача:

Сложить два числа: $A_{10} = +16$, $B_{10} = -7$ в ОК (обратный код) и ДК (дополнительный код).

Решение:

Требуется преобразование $A+(-B)$, в котором второй член записывается с учетом его знака:

$$[A_2]_{\Pi} = [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0|10000;$$

$$[B_2]_{\Pi} = 1|111 = 1|00111; [B_2]_{\text{ок}} = 1|11000; [B_2]_{\text{дк}} = 1|11001$$

Сложение в ОК	Сложение в ДК
$\begin{array}{r} [A_2]_{\text{ок}} = 0 10000 \\ + [B_2]_{\text{ок}} = 1 11000 \\ \hline 10 01000 \\ + \backslash \text{-----} 1 \\ \hline 0 01001 \\ C_2 = 0 01001 \\ C_{10} = +9 \end{array}$	$\begin{array}{r} [A_2]_{\text{дк}} = 0 10000 \\ + [B_2]_{\text{дк}} = 1 11001 \\ \hline 10 01001 \\ \hline C_2 = 0 01001 \\ C_{10} = +9 \end{array}$

При складывании чисел в обратном и дополнительном кодах получены переносы в знаковый разряд и из знакового разряда. В случае первом случае (обратный код) при переносе из знакового разряда необходимо дополнительно прибавить единицу младшего разряда. Во втором случае (дополнительный код) данный перенос игнорируется.

Ответ:

9

3. Задания**Задача:**

Дано два десятичных двузначных целых числа: $A = 78$, $B = 56$.
Вычислить $(A-B)_{\text{ок}}$, $(B-A)_{\text{дк}}$.

Решение:

Прямой код

$$78 = 0 | 1001110_2 \quad -78 = 1 | 1001110_2$$

$$56 = 0 | 0111000_2 \quad -56 = 1 | 0111000_2$$

Обратный код

$$78 = 0 | 1001110_2 \quad -78 = 1 | 0110001_2$$

$$56 = 0 | 0111000_2 \quad -56 = 1 | 1000111_2$$

Дополнительный код

$$78 = 0 | 1001110_2 \quad -78 = 1 | 1001111_2$$

56=0 | 0111000₂ -56=1 | 0111001₂

A+(-B) ОК:

0 | 1001110₊

1 | 1000111

0 | 0010110=22₁₀

B+(-A) ДК:

0 | 0111000

1 | 1001111

0 | 0010110 = -22₁₀

Ответ:

0 | 0010010 - ок

0 | 0000111- дк



22

-22

Тест 3

1.	<p>Задание:</p>
	<p><input checked="" type="checkbox"/> Для представления целого числа может применяться</p>
	<p>Ответ:</p>
	<p> <input checked="" type="checkbox"/> А) нормализованный или ненормализованный код <input checked="" type="checkbox"/> В) прямой, обратный или дополнительный код <input checked="" type="checkbox"/> С) естественный или экспоненциальный код <input checked="" type="checkbox"/> D) логарифмический и показательный код </p>
2.	<p>Задание:</p>
	<p><input checked="" type="checkbox"/> Положительное число</p>
	<p>Ответ:</p>
	<p> <input checked="" type="checkbox"/> А) выглядит одинаково только в прямом и обратном кодах <input checked="" type="checkbox"/> В) выглядит одинаково только в обратном и дополнительном кодах <input checked="" type="checkbox"/> С) выглядит одинаково в прямом, обратном и дополнительном кодах <input checked="" type="checkbox"/> D) выглядит различно в прямом, обратном и дополнительном кодах </p>
3.	<p>Задание:</p>
	<p><input checked="" type="checkbox"/> Если взять отрицательное число и инвертировать разряды кроме знакового, то получится</p>
	<p>Ответ:</p>
	<p> <input checked="" type="checkbox"/> А) обратный код <input checked="" type="checkbox"/> В) прямой код <input checked="" type="checkbox"/> С) дополнительный код <input checked="" type="checkbox"/> D) двоичный код </p>
4.	<p>Задание:</p>
	<p><input checked="" type="checkbox"/> Дополнительный код числа получается</p>
	<p>Ответ:</p>
	<p> <input checked="" type="checkbox"/> А) из обратного кода прибавлением единицы к младшему разряду без переноса в знаковый разряд <input checked="" type="checkbox"/> В) из обратного кода прибавлением единицы к младшему разряду с переносом в знаковый разряд <input checked="" type="checkbox"/> С) из прямого кода прибавлением единицы к младшему разряду без переноса в знаковый разряд <input checked="" type="checkbox"/> D) из прямого кода прибавлением единицы к младшему разряду с переносом в знаковый разряд </p>

5.	Задание:
	<input checked="" type="checkbox"/> Если к двоичному числу без знака добавить знаковый разряд то получится
	Ответ:
6.	<input checked="" type="checkbox"/> А) обратный код
	<input checked="" type="checkbox"/> В) прямой код
	<input type="checkbox"/> С) дополнительный код
7.	<input type="checkbox"/> Д) двоичный код
	Задание:
	<input checked="" type="checkbox"/> Число $X = 14_{10}$ в восьми разрядном двоичном дополнительном коде Равняется
8.	<input type="checkbox"/> 00001110
	Ответ:
	<input checked="" type="checkbox"/> А) 00001110 В) 0110010
9.	<input type="checkbox"/> С) 1110001 Д) нет верного ответа
	Задание:
	<input checked="" type="checkbox"/> Восьми разрядное двоичное число $X = (10001010)_2$, заданное в дополнительном коде в десятичной системе равняется
10.	<input type="checkbox"/> 0001001
	<input checked="" type="checkbox"/> 1110110
	Ответ:
11.	А) -10 В) +10 С) -117 Д) -118
	Задание:
	<input checked="" type="checkbox"/> Восьми разрядное двоичное число $X = (00100111)_2$ заданное в обратном коде в десятичной системе равняется
12.	Ответ:
	А) -39 В) +39 С) -88 Д) +88
	Задание:
13.	<input checked="" type="checkbox"/> Число $X = -63_{10}$ в прямом коде будет представлено как
	Ответ:
	А) 10111111 В) 00111111 С) 10011111 Д) 00011111
14.	Задание:
	<input checked="" type="checkbox"/> Укажите дополнительный код десятичного числа -103 (минус сто три) в 8 разрядном компьютерном представлении.
	Ответ: 10011001
15.	<input checked="" type="checkbox"/>

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.

1. Теоретический материал

Для математических расчетов с использованием стандартных математических функций требуется импортировать соответствующую библиотеку:

Import math

После импорта к функциям библиотеки можно обращаться следующим образом:

math.имя_функции(...)

В таблице представлен синтаксис и описание ключевых математических функций библиотеки **math** языка Python

Функция	Назначение
ceil(x)	Округляет число x до ближайшего большего целого (округление "вверх").
floor(x)	Округляет число x до ближайшего меньшего целого (округление "вниз").
fabs(x)	Принимает абсолютное значение (модуль) числа x .
exp(x)	Принимает значение e^x .
log(x[, b])	Если у функции один аргумент x , то функция принимает значение натурального логарифма x . При передаче двух аргументов, второй выступает в качестве основания логарифма.
pow(x, y)	Принимает значение x в степени y .
sqrt(x)	Принимает значение квадратного корня из x .
acos(x)	Принимает значение арккосинуса x в радианах.
asin(x)	Принимает значение арксинуса x в радианах.
atan(x)	Принимает значение арктангенса x в радианах.
cos(x)	Принимает значение косинуса x , где x выражен в радианах.
sin(x)	Принимает значение синуса x , где x выражен в радианах.

tan(x)	Принимает значение тангенса 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):		
	<pre>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)</pre>	
Задача:		
	Для введенных чисел x и y найти значение функции	
	$\sin(xe^y), xy \leq -1$	
	$f(x,y) = \{\sqrt{ \cos(xy) }, -1 < xy < 5$	
	$x^2 + \operatorname{tgy}, xy \geq 5$	
Решение (код программы):		
	<pre>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)</pre>	
Задача:		
	Вычислить значение функции $f(x) = \sin(x - e^2) + 3x$ на отрезке $[x_n, x_k]$ с шагом h_x	

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

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

При этом x изменяется в отрезке $0 \leq x \leq 1$ с шагом $hx = 0.2$; y изменяется в отрезке $1 \leq y \leq 2$ 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))
        y = y + hy
    x = x + hx
```

```
#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

$$f(x,y) = \ln|\sin(x+y)|$$

Решение (код программы):

```
import math
x, y= int(input()), int(input())
def f(x,y):
    return math.log(abs(math.sin(x+y)), math.e)
print(f(x,y))
```

2. Задача:

Для введенных чисел x и y найти значение функции

$$f(x,y) = \begin{cases} \arctg \sqrt[3]{|x-y|}(xe^y), \sin(x+y) \leq -0,5 \\ 3\log_3(|xy|), -0,5 < \sin(x+y) < 0,5 \\ x^3 + y^{1,5}, \sin(x+y) \geq 0,5 \end{cases}$$

Решение (код программы):

```
import math
x, y= int(input()), int(input())
def f(x,y):
    if math.sin(x+y)<=-0.5:
        return math.atan(math.pow(abs(x-y),1/3))*(x*(math.e)**y)
    if -0.5<math.sin(x+y)<0.5:
        return 3*math.log(abs(x*y),3)
    if math.sin(x+y)>=0.5:
        return x**3+y**(3/2)
print(f(x,y))
```

3. Задача:

Вычислить значение функции $f(x) = \cos^3(e*x) + \sin|x|$ на отрезке $[a, b]$ с шагом hx

Решение (код программы):

```
import math
a, b, h= int(input()), int(input()), int(input())
def f(x):
    return (math.cos(math.e*x))**3 + math.sin(abs(x))
for x in range(a, b + 1, h):
    print(f(x))
```



4. **Задача:**

Вычислить значения функции

$$f(x,y) = \begin{cases} \sqrt[3]{\sin(xe^{0.1y})}, & \text{при } x + y \leq 2; \\ |\log_2(x + y)|, & \text{при } x + y > 2. \end{cases}$$

При этом x изменяется в отрезке $1 \leq x \leq 2.5$ с шагом $hx = 0.5$; y изменяется в отрезке $1 \leq y \leq 4$ с шагом $hy = 1$.

Решение (код программы):

```
import math
x = 1
hx = 0.5
y = 1
hy = 1
def f(x, y):
    if x + y <= 2: return math.sin(x*math.e**(0.1*y))**(1/3)
    else: return abs(math.log(x+y, 2))
for xn in range(4):
    x = x + hx * xn
    for yn in range(4):
        y = y + hy * yn
        print(f(x,y))
```