**Дисциплина «Информатика»**

**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 беззнаковых целых чисел наиболее удобен: битовый набор, соответствующий записи этого числа в двоичной системе счисления. Под целые числа без знака обычно отводится*k* = 8,16,32или64 разряда.  Таким образом, для получения компьютерного представления беззнакового целого числа достаточно перевести число в двоичную систему счисления и дополнить полученный результат слева нулями до стандартной разрядности. | |
| **2. Пример** | |
| ***Задача:*** | |
|  | Найти представление беззнакового числа 2610в восьмиразрядном битовом наборе |
| ***Решение:*** | |
|  | Переведем число 26 в двоичную систему счисления.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 26 | 2 |  |  |  |  | | -26 | 13 | 2 |  |  |  | | 0 | -12 | 6 | 2 |  |  | |  | 1 | -6 | 3 | 2 |  | |  |  | 0 | -2 | 1 |  | |  |  |  | 1 |  |  |   Результат перевода: 2610= 110102  Дополним полученный результат слева нулями до восьми шестнадцати  2610= **000**110102 |
| ***Ответ:*** | |
|  | **00011010** |

|  |  |  |  |
| --- | --- | --- | --- |
| **3. Задания** | | | |
| 1. | | ***Задача:*** | |
|  | Найти представление беззнакового числа 13210в шестнадцатиразрядном битовом наборе |
| ***Решение:*** | |
|  | 10000100 – в двоичной  0000000010000100 - в шестнадцатиразрядном битовом наборе |
| ***Ответ:*** | |
|  | 0000000010000100 |
| 2. | | ***Задача:*** | |
|  | Найти минимальное и максимальное значения чисел для 16-ти разрядного беззнакового представления |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 0; 65535 |
| **1. Теоретический материал** | | | |
| Для представления знаковых целых чисел используются три способа:   1. прямой код; 2. обратный код; 3. дополнительный код.   Все три способа используют самый левый (старший) разряд битового набора длины *k* для кодирования знака числа: знак “плюс” кодируется нулем, а “минус” – единицей. Остальные *k*-1 разрядов (называемые *мантиссой* или цифровой частью) используются для представления абсолютной величины числа.  Положительные числа в **прямом, обратном и дополнительном кодах** изображаются одинаково – цифровая часть содержит двоичную запись числа, в знаковом разряде содержится 0.  Для представления отрицательного числа в **прямом коде**, в знаковый разряд помещается цифра 1, а в разряды цифровой части числа – двоичный код его абсолютной величины.  **Обратный код** отрицательного числаполучается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы – нулями  **Дополнительный код** отрицательных чисел получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. | | | |
| **2. Пример** | | | |
| ***Задача:*** | | | |
|  | Перевести число 45 в прямой, обратный и дополнительный код (*k* = 8) | | |
| ***Решение:*** | | | |
|  | Сначала переведем десятичное число **45** в двоичную систему счисления.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 45 | 2 |  |  |  |  |  | | -44 | 22 | 2 |  |  |  |  | | **1** | -22 | 11 | 2 |  |  |  | |  | **0** | -10 | 5 | 2 |  |  | |  |  | **1** | -4 | 2 | 2 |  | |  |  |  | **1** | -2 | **1** |  | |  |  |  |  | **0** |  |  |   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 **-5610**. | | |
| ***Решение:*** | | | |
|  | Выполним перевод положительного числа **56** в двоичную систему счисления, получим: **5610 = 1110002**.  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:*** | |
|  | 1100 – двоичной  0,0001100 – прямой  0,0001100 – обратный  0,0001100 - дополнительный |
| ***Ответ:*** | |
|  | 0,0001100 – прямой  0,0001100 – обратный  0,0001100 - дополнительный |
| 2. | | ***Задача:*** | |
|  | Найти прямой, обратный и дополнительный коды в однобайтовом представлении для числа **-3510**. |
| ***Решение:*** | |
|  | 100011  1,0100011 – прямой  1,1011100 - обратный  1,1011101 - дополнительный |
| ***Ответ:*** | |
|  | 1,0100011 – прямой  1,1011100 - обратный  1,1011101 - дополнительный |
| 3. | | ***Задача:*** | |
|  | Задан дополнительный код числа в однобайтовом представлении: **1,1011100**. Найти число в десятичной системе счисления. |
| ***Решение:*** | |
|  | 1,1011011 – обратный  1,0100100 - прямой  100100 - двоичный  -36 – в десятичной |
| ***Ответ:*** | |
|  | -36 |
| **1. Теоретический материал** | | | |
| Сложение и вычитание беззнаковых чисел происходит по обычным для позиционных систем счисления алгоритмам.  Сложение в обратном коде происходит следующим образом: по-обычному алгоритму складываются все разряды, включая знаковый. Результат такого сложения для *k*-разрядных наборов имеет длину *k*+1 (самый левый разряд результата равен единице, если был перенос при сложении старших разрядов операндов, иначе – нулю).Значение левого *k* +1-го разряда добавляется к младшему разряду результата. Получаем *k*-разрядный набор, который и будет суммой двух чисел в обратном коде.Вычитание чисел в обратном коде *x* – *y* сводится к сложению *x+* (*–y*).  В дополнительном коде сложение происходит так: по обычному алгоритму складываются все разряды, включая знаковый; единица переноса в *k* +1-й разряд отбрасывается. | | | |
| **2. Пример** | | | |
| ***Задача:*** | | | |
|  | Сложить два числа: А10 = 7, В10 = 16. | | |
| ***Решение:*** | | | |
|  | Переведем числа в двоичную систему счисления  А2 = +111 = +0111; В2 = +10000.  Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:  [A2]п = [A2]ок = [A2]дк = 0|00111; [В2]п = [В2]ок = [В2]дк = 0|10000.  Сложение в обратном или дополнительном коде дает один и тот же результат:  0|00111  +0|10000  ----------  С2 = 0|10111  С10 = +23 | | |
| ***Ответ:*** | | | |
|  | 23 | | |
| ***Задача:*** | | | |
|  | Сложить два числа: А10= +16, В10= -7 в ОК (обратный код) и ДК (дополнительный код). | | |
| ***Решение:*** | | | |
|  | Необходимо преобразование А+(-В), в которой второй член преобразуется с учетом знака  [A2]п = [A2]ок = [A2]дк = 0|10000;  [В2]п = 1|111 = 1|00111; [В2]ок = 1|11000; [В2]дк = 1|11001    При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового разряда. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда. В случае ДК этот перенос игнорируется. | | |
| ***Ответ:*** | | | |
|  | **9** | | |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| ***Задача:*** | |
|  | Дано два десятичных двузначных целых числа: А = 78, В = 56. Вычислить (А-В)ок, (В-А)дк. |
| ***Решение:*** | |
|  | 78 – 1001110 - двоичный  56 - 111000 – двоичный  +A обратный – 0,1001110  -A обратный – 1,0110001  +В обратный – 0,0111000  -B обратный – 1,1000111   1. B = 0,0010110 = +22 – в десятичной   B-A = 1,1101010 = -22 – в десятичной |
| ***Ответ:*** | |
|  | +22;-22 |

|  |  |  |
| --- | --- | --- |
| **Тест** | | |
| **1.** | ***Задание:*** | |
|  | Для представления целого числа может применяться |
| ***Ответ:*** | |
|  | A) нормализованный или ненормализованный код  B) прямой, обратный или дополнительный код  C) естественный или экспоненциальный код  C) логарифмический и показательный код |
| **2.** | ***Задание:*** | |
|  | Положительное число |
| ***Ответ:*** | |
|  | A) выглядит одинаково только в прямом и обратном кодах  B) выглядит одинаково только в обратном и дополнительном кодах C) выглядит одинаково в прямом, обратном и дополнительном кодах  D) выглядит различно в прямом, обратном и дополнительном кодах |
| **3.** | ***Задание:*** | |
|  | Если взять положительное число и все разряды, включая знаковый разряд инвертировать, то получится |
| ***Ответ:*** | |
|  | A) обратный код  B) прямой код  C) дополнительный код  D) двоичный код |
| **4.** | ***Задание:*** | |
|  | Дополнительный код числа получается |
| ***Ответ:*** | |
|  | A)из обратного кода прибавлением единицы к младшему разряду без переноса в знаковый разряд  B) из обратного кода прибавлением единицы к младшему разряду с переносом в знаковый разряд  C) из прямого кода прибавлением единицы к младшему разряду без переноса в знаковый разряд  D) из прямого кода прибавлением единицы к младшему разряду с переносом в знаковый разряд |
| **5** | ***Задание:*** | |
|  | Если к двоичному числу без знака добавить знаковый разряд то получится |
| ***Ответ:*** | |
|  | A) обратный код  B) прямой код  C) дополнительный код  D) двоичный код |
| **6.** | ***Задание:*** | |
|  | Число Х=1410 в восьми разрядном двоичном дополнительном коде равняется |
| ***Ответ:*** | |
|  | A) 0001110 B) 1110001C) 1110001D) нет верного ответа |
| **7.** | ***Задание:*** | |
|  | Восьми разрядное двоичное число Х=(10001010)2, заданное в дополнительном коде в десятичной системе равняется |
| ***Ответ:*** | |
|  | A) –10 B) +10C) –117 D) –118 |
| **8.** | ***Задание:*** | |
|  | Восьми разрядное двоичное число Х=(00100111)2 заданное в обратном коде в десятичной системе равняется |
| ***Ответ:*** | |
|  | **A)** –39 B) +39 C) –88D)+88 |
| **9** | ***Задание:*** | |
|  | Число Х= -6310 в прямом коде будет представлено как |
| ***Ответ:*** | |
|  | A) 10111111B)00111111 C) 10011111 D) 00011111 |
| **10** | ***Задание:*** | |
|  | Найти прямой, обратный и дополнительный код десятичного числа -103в 8разрядном компьютерном представлении. |
| ***Ответ:*** | |
|  | 1,1100111 - прямой  1,0011000 – обратный  1,0011001 - дополнительный |

**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 работы с математическими функциями нужно импортировать библиотеку**math:**  importmath  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*ex*. | | 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)* = 2*yx*+ ln|*x*+y3| |
| ***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 |
| ***Решение (кодпрограммы):*** | |
|  | 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* – e2) + 3*x* на отрезке [*xn*, *xk*] с шагом *hx* |
| ***Решение (кодпрограммы):*** | |
|  | import math  xn = float(input('Введитеxn '))  xk = float(input('Введитеxk '))  hx = float(input('Введитеhx '))  x = xn #устанавливаем xв начало отрезка в xn  whilex<= 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    При этом*x* изменяется в отрезке  с шагом;*y* изменяется в отрезке с шагом. |
| ***Решение (кодпрограммы):*** | |
|  | import math  xn = float(input('Введитеxn '))  xk = float(input('Введитеxk '))  hx = float(input('Введитеhx '))  yn = float(input('Введитеyn '))  yk = float(input('Введитеyk '))  hy = float(input('Введитеhy '))  x = xn #устанавливаем xв начало отрезка в xn  whilex<= xk:#пока не дойдем до xk  y = yn#устанавливаем yв начало отрезка в yn  whiley<= yk: #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  y = y + hy#прибавляем к yшаг  x = x + hx#прибавляем к xшаг |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Для введенных чисел *x*и *y*найти значение функции |
| ***Решение(кодпрограммы):*** | |
|  | import math as m  x = float(input()) y = float(input()) func = m.log(m.fabs(m.sin(x + y))) print(func) |
| 2. | ***Задача:*** | |
|  | Для введенных чисел *x*и *y*найти значение функции |
| ***Решение (код программы):*** | |
|  | import math as m  x = float(input()) y = float(input()) if m.sin(x + y) <= -0.5:  func = m.atan(m.pow((m.fabs(x - y)), 1 / 3)) \* (x \* m.pow(m.e, y)) elif m.sin(x + y) < 0.5:  func = 3 \* m.log(m.fabs(x \* y), 3) - 0.5 elif m.sin(x + y) >= 0.5:  func = x \*\* 3 + y \*\* 1.5 print(func) |
| 3. | ***Задача:*** | |
|  | Вычислить значение функции *f*(*x*) = cos3(e*x*) + sin|x| на отрезке [*xn*, *xk*] с шагом *hx* |
| ***Решение (код программы):*** | |
|  | import math as m  def func(x):  f = m.cos(m.e \* x) \*\* 3 + m.sin(abs(x))  return f  xn,xk,hx=float(input()),float(input()),float(input()) x=xn while x<=xk:  print(func(x))  x+=hx |
| 4. | ***Задача:*** | |
|  | Вычислить значения функции    При этом *x* изменяется в отрезке  с шагом ; *y* изменяется в отрезке с шагом . |
| ***Решение (код программы):*** | |
|  | import math as m   def func(x, y):  if x + y <= 4:  f = (m.sin(x \* (m.e \*\* (0.1 \* y)))) \*\* (1 / 3)  elif x + y > 2:  f = abs(m.log(x + y, 2))  return f   x = 1 while x <= 2.5:  y = 1  while y <= 4:  print(func(x, y))  y += 1  x += 0.5 |