Лабораторная работа 4

Тема. Битовые операции над структурами данных.

**Цель.**

* Получить знания по операциям языка программирования С++ для обработки битовой информации.
* Получение практических навыков по выполнению операций над битовым представлением данных

**Задание.**

1. Разработать программу, которая реализует задачу варианта, используя в операциях внутреннее представление данных – их двоичный код.
2. Требования.

Выполнить декомпозицию задачи

Выделенные подзадачи оформить в виде подпрограмм.

ВАРИАНТЫ

|  |  |
| --- | --- |
| Номер Варианта | Задание варианта |
|  | Разработать программу, которая моделирует операцию сложения двух целых положительных числа (как ее выполняет компьютер), используя их внутреннее двоичное представление. |
|  | Разработать программу, которая моделирует операцию вычитания двух целых положительных числа как ее выполняет компьютер), используя их внутреннее двоичное представление. |
|  | Разработать программу, которая переводит число из двоичной системы счисления (из внутреннего представления) в десятичную систему счисления. |
|  | Разработать программу, которая переводит число из двоичной системы счисления в восьмеричную. |
|  | Разработать программу, которая переводит число из двоичной системы счисления (из внутреннего представления) в шестнадцатеричную |
|  | Разработать программу, которая создает множество из произвольного числа элементов, целого типа. Множество представить на битовом векторе. |
|  | Разработать программу, которая выводит двоичный код целого или вещественного числа. |
|  | Разработать программу, которая выполняет сортировку большого числа элементов за минимальное время, используя битовый массив. |
|  | Разработать программу, которая создает множество из произвольного числа элементов, целого типа, используя для хранения множества битовый вектор, и выполняет операцию объединения двух множеств. |
|  | Разработать программу, которая создает множество из произвольного числа элементов, целого типа, используя для хранения множества битовый вектор, и выполняет операцию пересечения двух множеств. |
|  | Разработать программу шифрования текста, рассматривая текст как последовательность битов. Суть алгоритма шифрования: имеется матрица размером 8\*8 первоначального преобразования битовой последовательности, ячейки которой хранят номера битов входной последовательности. Необходимо создать выходную битовую последовательность, располагая значения битов согласно их номеров в матрице.  Пояснение. Пусть матрица 3\*3 содержит значения  9 3 5  8 2 4  7 1 6  Следовательно, в выходной битовой последовательности на первом месте будет значение бита с номером 9 из входной последовательности, на втором месте бит с номером 3, на третьем бит с номером 5, на четвертом с номером 8 и т.д. |
|  | Разработать программу, которая обрабатывает данные, снятые с выходов нескольких устройств и формирует список значений в десятичной системе, снятых с каждого устройства. Данные каждого устройства представляют последовательность из нескольких битов. Известно, что с устройства У1 снято 35 битов, с устройства У2 – 18 битов и с У3-12 битов. Во входной последовательности сначала идут данные с У1, затем с У2 и затем с У3. |
|  | Разработать программу, которая определяет количество единиц в двоичном представлении числа любого типа. Для вещественного типа использовать структур данных union. |
|  | Разработать программу, которая моделирует операцию вычитания двух целых чисел. |
|  | Разработать программу, которая в последовательности целых чисел находит те, последний байт которых содержит двоичный код, являющийся зеркальным отражением двоичного кода первого байта числа. |
|  | Дано целое число A и целое неотрицательное число k. Обнулите у числа A его последние k бит и выведите результат. |
|  | Дано целое число A и целое неотрицательное число k. Выведите число, которое получается из числа A инвертированием k-го бита. |
|  | Дано целое число A и натуральное число k. Выведите число, которое состоит только из k последних бит числа A (то есть обнулите все биты числа A, кроме последних k). В этой задаче можно использовать сложение и вычитание. |
|  | Даны числа *a* и *b*. Используя только битовые операции и операции сложения и вычитания, вычислите число *x*=(36*a*+[*b*16]) % 32. Выведите результат на экран. |
|  | Алгоритм шифрования перемешиванием:  Дано число, переставьте его соседние биты (то есть поменяйте местами биты с номерами 0 и 1, 2 и 3, 4 и 5 и т.д.). Разрешается использовать битовые операции. Запрещается использовать арифметические операции, ветвления, циклы. Общее число бит в числе не превосходит 32. (Например, для числа 78 результат 141). |
|  | Алгоритм хеширования. Используется 32-битная арифметика. Переменная *h* хранит текущее значение хеш-функции (h(key) – это хеш-функция, где key ‘это последовательность символов – ключ доступа к данным). Далее для каждого считанного байта key сообщения выполняются следующие операции:  1. Значение *h* сдвигается на 4 бита влево, к нему прибавляется (арифметическим суммированием) значение очередного символа ключа.  2. Если хотя бы один из 4 старших битов *h* равен 1, то старшие 4 бита сдвигаются на 24 бита вправо, и выполняется операция побитового исключающего ИЛИ со значением *h*. После чего обнуляются старшие 4 бита значения *h*.  Все операции проводятся с 32-битными числами, то есть берутся 32 младших бита результата. Например, для ключа AND результат 17956. |
|  | Контрольная сумма хранится в двубайтовой целочисленной переменной *h*. Инициализированная значением 0. Контрольная сумма формируется из символов поступающего текста. Текст храниться в переменной С.  Далее с каждым считанным байтом *С* выполняются следующие операции:   * Значение *h* циклически сдвигается вправо (то есть последний бит становится первым, не забываем, что число *h* является 16-битным). * К значению *h* прибавляется значение считанного байта (то есть ASCII-кода его), от результата берется последние 16 бит.   Вот пример вычисления контрольной суммы для строки “AND“  h = 0b0000000000000000 - циклический сдвиг вправо  h = 0b0000000001000001 - добавили 65 - ASCII-код A  h = 0b1000000000100000 - циклический сдвиг вправо  h = 0b1000000001101110 - добавили 78 - ASCII-код N  h = 0b0100000000110111 - циклический сдвиг вправо  h = 0b0100000001111011 - добавили 68 - ASCII-код D  Результат 16 507 |