Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт

по лабораторной работе № 5

на тему:

“Битовые операции”

по дисциплине “Программирование. Дополнительные главы”

Выполнил: студент гр. 4306 Табаков А.В.

Принял: к.т.н., доцент Сискович Т.И.

Санкт-Петербург  
2015 г.

**Цель**

Получить практические навыки работы с битовыми операциями.

**Задание**

Написать программу для: циклического сдвига битов в мантиссе между 19 и 11 битами.

**Уточнение задания**

В программе должно быть использовано простейшее меню. Выполнение программы должно быть многократным по желанию пользователя. Пользователь вводит число с плавающей точкой и количество сдвигов. Программа должна совершать циклический сдвиг влево с 19 по 11 бит, нумерация битов начинается с нуля.

**Описание структуры**

Для решения задач разработана структура:

typedef union sFLOAT{

float asFloat; //число с плавающей точкой

long int asInteger; //число с плавающей точкой как длинное целое

}MyFLOAT;

**Контрольные примеры**

Контрольные примеры представлены на рисунке 1*.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходные данные | | | | | Результат | | | |
| Число | Кол-во сдвигов | Знак | Порядок | Двоичное представление мантиссы | Число | Знак | Порядок | Двоичное представление мантиссы |
| 0.15625 | 3 | 0 | 01111100 | 101 000000000 000000000000 | 0.156250 | 0 | 01111100 | 101 000000000 000000000000 |
| 234.125 | 4 | 0 | 10000110 | 111 010100010 000000000000 | 226.625 | 0 | 10000110 | 111 000101010 000000000000 |
| 62.245201 | 2934 | 0 | 10000100 | 111 110001111 101000000000 | 63.557701 | 0 | 10000100 | 111 211100011 101000000000 |

Рис. 1. Контрольные примеры

**Описание главной функции**

Назначение: организация управления порядком вызова функций.

**Описание переменных функции**

Описание переменныхпредставлено вТаблице 1.

Таблица 1*.*Описание переменных главной функции

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| source | MyFLOAT | Исходное число, записанное в объединение |
| count | int | Количество сдвигов |
| processed | MyFLOAT | Сформированное число, записанное в объединение |
| Q | Int | Переменная выбора меню |
| init | Bool | Флаг инициализации (true – введены; false – не введены) |
| process | Bool | Флаг обработки данных (true – обработаны; false – не обработаны) |

**Описание функций**

**Описание функции output**

Назначение: вывод мантиссы в двоичном виде.

Прототип: void output(MyFLOAT source, MyFLOAT processed, int count, int key); описание формальных переменных представлено на рисунке 2.

Пример вызова: output(source, processed, count, 0);

Вызывающая функция: main.

**Описание переменных**

Описание переменных функции output представлено на рисунке 2.

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| Формальные переменные | | |
| source | MyFLOAT | Исходное число, записанное в объединение |
| processed | MyFLOAT | Сформированное число, записанное в объединение |
| count | int | Количество сдвигов |
| key | int | Ключ вывода (0 - вывод только исходного числа, 1 вывод всех) |
| Локальные переменные | | |
| i | int | Вспомогательная переменная |

Рис. 2.Описание переменных функции output

**Описание функции menu**

Назначение: вывод меню программы.

Прототип: int menu();

Возвращаемое значение: номер пункта меню.

Пример вызова: Q=menu();

Вызывающая функция: main.

**Описание переменных**

Описание переменных функции menu представлено на рисунке 3.

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| Локальные переменные | | |
| Q | int | Переменная выбора пункта меню |

Рис. 3.Описание переменных функции menu

**Описание функции messages**

Назначение: Функция используется для ввода сообщений пользователю.

Прототип: void messages(int); описание формальных переменных представлено на рисунке 4.

Возвращаемое значение: int номер вызываемого сообщения.

Пример вызова: messages(1);

Вызывающая функция: main.

Сообщения:

messages(1): "Число введено успешно"

messages(2): "Исходное число не введено, выберите 1 пункт меню для ввода"

messages(3): "Число не обработано, выберите 4 пункт меню для обработки"

messages(4): "До новых встреч!"

messages(5): "Число успешно обработано"

**Описание переменных**

Описание переменных функции messages представлено на рисунке 4.

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| Формальные переменные | | |
| Key | int | Вспомогательная переменная |

Рис. 4. Описание переменных функции messages

**Описание функции enterNum**

Назначение: ввод чисел в заданном диапазоне.

Прототип: int enterNum(int first, int last); описание формальных переменных представлено на рис.5.

Возвращаемое значение: целое число.

Пример вызова: Q=enterNum(1, 7);

Вызывающая функция: main.

**Описание переменных**

Описание переменных функции enterNum представлено на рисунке 5.

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| Локальные переменные | | |
| Num | int | Вспомогательная переменная |
| check\_num | bool | Флаг является ли символ цифрой |
| check\_all | bool | Флаг является ли строка числом |
| Str | char\* | Вспомогательная переменная |
| Формальные переменные | | |
| First | int | Начальное число |
| Last | int | Конечное число |

Рис. 5*.*Описание переменных функции enterNum

**Описание функции leftShift**

Назначение: циклический битовый сдвиг с 19 по 11 биты.

Прототип: long int leftShift(long int num, int k); описание формальных переменных представлено на рисунке 6.

Возвращаемое значение: целое длинное число со сдвинутыми в мантиссе битами на k позиций.

Пример вызова: processed.asInteger=leftShift(source.asInteger, count); описание фактических переменных представлено в таблице 1.

Вызывающая функция: main.

**Описание переменных**

Описание переменных функции leftShift представлено на рисунке 6.

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| Формальные переменные | | |
| N | long int | Исходное число |
| k | int | Количество сдвигов |

Рис. 6*.*Описание переменных функции leftShift

**Структура вызова функций**

Структура вызова функций представлена на рисунке 7.

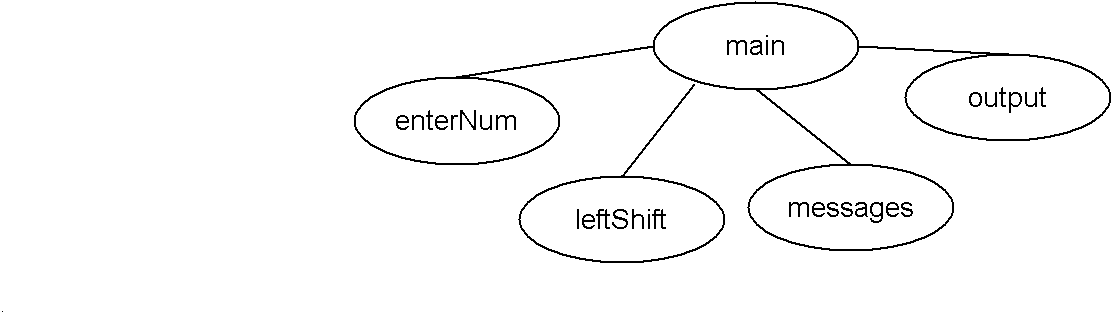
****

Рис. 7. Структура вызова функций

**Текст программы с комментариями**

#include <stdlib.h>

#include <locale.h>

#include <windows.h>

#include <conio.h>

typedef union sFLOAT{

float asFloat;

long int asInteger;

}MyFLOAT;

int menu(); //Прототип функции Главного меню

int enterNum(int first, int last); //Прототип функции ввода целочисленных значений

//в диапазоне

void output(MyFLOAT source, MyFLOAT processed, int count, int key); //Прототип функции вывода мантиссы в двоичном

//виде

long int leftShift(long int num, int k); //Прототип функции циклического сдвига

void messages(int Key); //Прототип функции вывода сообщения

int main()

{

system("mode con cols=80 lines=20");

MyFLOAT source, processed;

int Q, count;

bool init=false, processing=false;

system("chcp 1251");

do

{

switch (Q=menu())

{

case 0:

puts("Программа делает циклический сдвиг влево в мантиссе числа");

puts("с 19 по 11 биты, отсчёт битов начинается с нуля, область изменённых битов отделена пробелами");

break;

case 1:

system("cls");

puts("Введите число с плавающей точкой, точность записи: 6 разрядов после точки");

puts("Число вводится в формате X.Y - где X,Y произвольные числа");

puts("Важно! Если ввести запятую вместо точки, числа после запятой теряются");

printf("Введите число - ");

scanf("%f", &source.asFloat);

fflush(stdin);

printf("Введите число сдвигов от 0 до 32767 - ");

count=enterNum(-1, INT\_MAX);

count=count%8;

init=true;

processing=false;

messages(1);

break;

case 2:

if(init)

{

system("cls");

output(source, processed, count, 0);

system("pause");

}

else

messages(2);

break;

case 3:

if(init)

{

processed.asInteger=leftShift(source.asInteger, count);

processing=true;

messages(5);

}

else

messages(2);

break;

case 4:

if(init)

if(processing)

{

system("cls");

output(source, processed, count, 1);

system("pause");

}

else

messages(3);

else

messages(2);

break;

case 5:

messages(4);

break;

default:

messages(7);

}

}

while (Q!=5);

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция меню

int menu()

{

int Q;

system("cls");

puts("Главное меню");

puts("0 - Справка");

puts("1 - Ввод числа");

puts("2 - Вывод исходных данных");

puts("3 - Обработка");

puts("4 - Вывод результата");

puts("5 - Выход");

printf("Введите номер пункта - ");

scanf("%d", &Q);

fflush(stdin);

return Q;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция вывода символа и кода символа

void output(MyFLOAT source, MyFLOAT processed, int count, int key)

{

printf("Количество сдвигов - %d\n", count);

printf("Исходное число - %.6f\n", source.asFloat);

printf("Следующие числа в двоичном представлении\n");

printf("Знак - %d, Порядок - ", (source.asInteger >> 31) & 0x00000001);

for(int i = 30; i >= 23; i--)

printf("%ld", source.asInteger >> i & 0x01);

printf(" Мантисса - ");

for (int i = 22; i >= 0; i--)

{

if(i==19 || i == 11)

printf(" ");

printf("%ld", source.asInteger >> i & 0x000001);

}

if(key)

{

puts("");

printf("Сформированное число - %.6f\n", processed.asFloat);

printf("Следующие числа в двоичном представлении\n");

printf("Знак - %d, Порядок - ", (processed.asInteger >> 31) & 0x00000001);

for(int i = 30; i >= 23; i--)

printf("%ld", processed.asInteger >> i & 0x01);

printf(" Мантисса - ");

for (int i = 22; i >= 0; i--)

{

if(i==19 || i == 11)

printf(" ");

printf("%ld", processed.asInteger >> i & 0x000001);

}

}

puts("");

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция обработки

long int leftShift(long int n, int k)

{

return (((n&0xFF000) << k | n >> (8 - k)) & 0xFF000) | (n & 0xFFF00FFF);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция ввода целочисленных переменных в диапазоне

int enterNum(int first, int last)

{

int num;

bool check\_num, check\_all;

char str[4];

const char numbers[]="0123456789";

do

{

check\_all=true;

check\_num=false;

scanf("%s", &str);

fflush(stdin);

for(int i=0; str[i]!='\0' && check\_all; i++)

{

for(int j=0; numbers[j]!='\0' && !check\_num; j++)

if(str[i]==numbers[j] || str[i]=='\0')

check\_num=true;

if(check\_num)

check\_num=false;

else

check\_all=false;

}

if(check\_all)

num=atoi(str);

else

printf("В строку попало что-то кроме числа, повторите ввод:\n");

if((num < first || num > last) && check\_all)

printf("Возможно вы ошиблись при вводе?\nВведите число от %d до %d\nПовторите ввод: ", first, last);

}

while(num < first || num > last || !check\_all);

return num;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Функция вывода сообщений пользователю

void messages(int Key)

{

system("cls");

switch(Key)

{

case 1:

puts("Данные введёны успешно");

break;

case 2:

puts("Исходные данные не введены, выберите 1 пункт меню для ввода");

break;

case 3:

puts("Мантисса не обработана, выберите 3 пункт меню для обработки");

break;

case 4:

puts("До новых встреч!");

break;

case 5:

puts("Данные обработаны успешно");

break;

case 7:

puts("Что-то пошло не так, повтороите ввод");

break;

}

system("pause");

}

**Результаты решения задачи**

При выполнении программы были получены результаты, совпадающие со значениями, приведенными на рисунке 1. Ошибок не обнаружено.

**Вывод**

При выполнении лабораторной работы были получены практические навыки работы c битовыми операциями на языке программирования «С/C++».