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

Отчет по лабораторной работе №1

по дисциплине “Организация ЭВМ и систем”

на тему: “Исследование внутреннего представления различных форматов данных”

Выполнил: Хамитов А.К. 9308 Принял: Жандаров В.В.

# Содержание

[Содержание 2](#_Toc52350510)

[1. Задание 3](#_Toc52350511)

[2. Блок-схема алгоритма 4](#_Toc52350512)

[3. Листинг программы 5](#_Toc52350513)

[4. Примеры запуска программ 6](#_Toc52350514)

[5. Структурная схема аппаратных средств 7](#_Toc52350515)

# 1. Задание

Общее задание:

1. В зависимости от номера варианта задания разработать алгоритм ввода с клавиатуры требуемых типов данных и показать на экране их внутреннее представление в двоичной системе счисления.

2. Написать и отладить программу на языке С++, реализующую разработанный алгоритм.

3. В соответствии с заданием дополнить разработанный ранее алгоритм блоками для выполнения преобразования двоичного полученного кода исходного типа данных и последующего вывода преобразованного кода в двоичной системе счисления и в формате исходного данного.

Вариант 24(5)

Поменять местами заданные пользователем группы рядом стоящих бит, номера старших разрядов этих групп и количество бит в группе, вводится с клавиатуры.

# 2. Блок-схема алгоритма

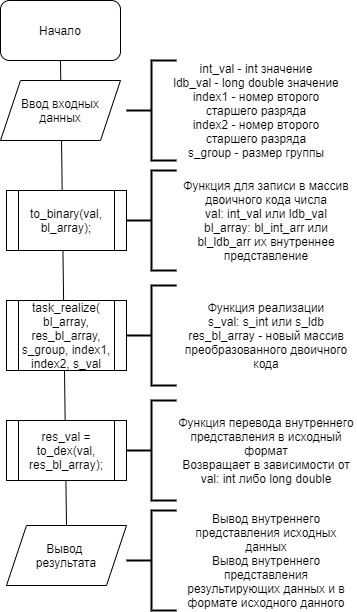


Рисунок 1. Блок-схема алгоритма main.

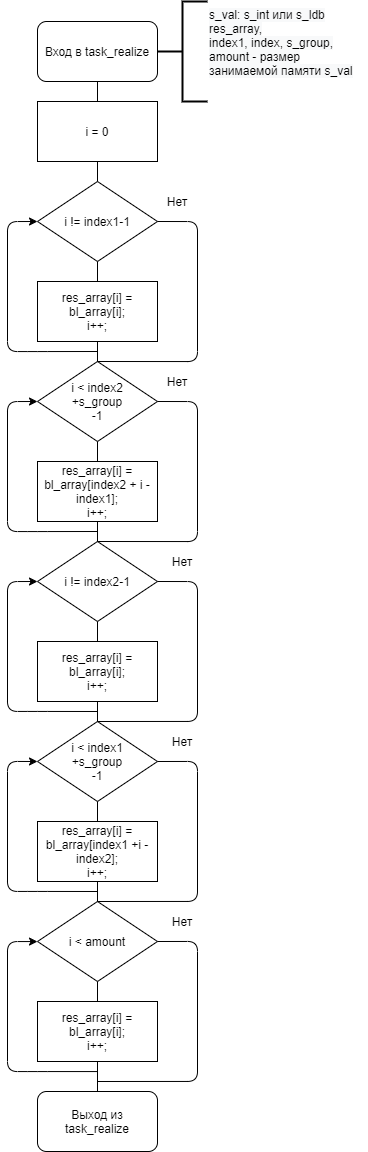


Рисунок 2. Блок-схема алгоритма handle

# 3. Листинг программы

/\*

    Хамитов Абулкаир 9308

    Вариант 24(4)

    Задача:

    int/long double

    Преобразование:

    Поменять местами заданные пользователем группы рядом стоящих бит, номера старших разрядов этих групп и количество бит в группе, вводится с клавиатуры.

\*/

#include <iostream>

using namespace std;

#ifdef \_\_linux\_\_

    #define CLEAR "clear"

#else

    #define CLEAR "cls"

#endif

// ----- Константы ------

const int s\_int = 32;

const int s\_ldb = 80;

// ----- Прототипы функций -----

// Функции конвертирования

void to\_binary(long double, bool\*);

void to\_binary(int, bool\*);

int to\_dex(int, bool\*);

long double to\_dex(long double, bool\*);

// Функции реализации

void task\_realize(bool\*, bool\*, short, short, short, short);

// Декоративные функции

void print\_binary(bool\*, short);

int main()

{

    long double ldb\_val;    // Переменная для хранения long double

    int         int\_val;    // Переменная для хранения int

    short   menu,       // Номер меню

            index1,     // Номер первого старшего разряда

            index2,     // Номер второго старшего разряда

            s\_group;    // Размер группы

    bool bl\_int\_array[s\_int];   // Бинарный код int

    bool bl\_ldb\_array[s\_ldb];   // Бинарный код long double

    bool res\_bl\_int\_array[s\_int];   // Бинарный код int после преобразования

    bool res\_bl\_ldb\_array[s\_ldb];   // Бинарный код long double после преобразования

    do

    {

        // Меню

        cout << "\nMenu";

        cout << "\n1) Int input \n2) Long double input \n0) Exit" << endl;

        cin >> menu;

        system(CLEAR);

        switch(menu)

        {

            case 1:

                // int

                cout << "\nEnter your int value: ";

                cin >> int\_val;

                // Конвертирование и вывод

                to\_binary(int\_val, bl\_int\_array);

                print\_binary(bl\_int\_array, s\_int);

                // Ввод размера группы

                do {

                    cout << "\n\nEnter group size: ";

                    cin >> s\_group;

                    if(s\_group > s\_int) cout << "\nIncorrect input. Try again";

                } while (s\_group > s\_int);

                // Ввод старших разрядов группы

                do {

                    cout << "Enter the first senior position: ";

                    cin >> index1;

                    cout << "Enter the second senior position: ";

                    cin >> index2;

                    if(index1+1 > s\_int || index2+1 > s\_int) cout << "\nIncorrect input. Try again";

                } while (index1+1 > s\_int || index2+1 > s\_int);

                // Реализация и вывод

                task\_realize(bl\_int\_array, res\_bl\_int\_array, s\_group, index1, index2, s\_int);

                print\_binary(res\_bl\_int\_array, s\_int);

                cout << "\nDex    code: " << to\_dex(int\_val, res\_bl\_int\_array);

            break;

            case 2:

                // long double

                cout << "\nEnter your long double value: ";

                cin >> ldb\_val;

                // Конвертирование и вывод

                to\_binary(ldb\_val, bl\_ldb\_array);

                print\_binary(bl\_ldb\_array, s\_ldb);

                // Ввод размера группы

                do {

                    cout << "\n\nEnter group size: ";

                    cin >> s\_group;

                    if(s\_group > s\_ldb) cout << "\nIncorrect input. Try again";

                } while (s\_group > s\_ldb);

                // Ввод старших разрядов группы

                do {

                    cout << "Enter the first senior position: ";

                    cin >> index1;

                    cout << "Enter the second senior position: ";

                    cin >> index2;

                    if(index1+1 > s\_ldb || index2+1 > s\_ldb) cout << "\nIncorrect input. Try again";

                } while (index1+1 > s\_ldb || index2+1 > s\_ldb);

                // Реализация и вывод

                task\_realize(bl\_ldb\_array, res\_bl\_ldb\_array, s\_group, index1, index2, s\_ldb);

                print\_binary(res\_bl\_ldb\_array, s\_ldb);

                cout << "\nDex    code: " << to\_dex(ldb\_val, res\_bl\_ldb\_array);

            break;

            case 0:

                cout << "\nGood bye!";

            break;

            default:

                cout << "Incorrect input. Try again";

            break;

        }

    } while (menu != 0);

    return 0;

}

void to\_binary(long double val, bool \*bl\_array)

{

    int i, j, k;

    union {

        long double ss;

        unsigned char c[s\_ldb];

    } code;

    code.ss = val; //переменная объединения получила вещественное значение

    k = 0;

    for (j = s\_ldb/8 - 1; j >= 0; j--)

        for (i = 128; i; i >>= 1) //Разбираем каждый бит из байта

            bl\_array[k++] = i & code.c[j];

}

void to\_binary(int val, bool \*bl\_array)

{

    short k = 0;

    for(int i = s\_int-1; i >= 0; --i)

        bl\_array[k++] = ((val >> i) & 1);

}

long double to\_dex(long double val, bool \*new\_bl\_array)

{

    int i, j, k;

    union {

        long double ss;

        unsigned char c[s\_ldb];

    } code;

    code.ss = val;

    k = 0;

    for (j = s\_ldb/8 - 1; j >= 0; j--)

        for (i = 7; i >= 0; --i)

        // Меняем старый код long double на новый

            if(new\_bl\_array[k++])   code.c[j] |= (1<<i);

            else                    code.c[j] &=~(1<<i);

    return code.ss;

}

int to\_dex(int val, bool \*new\_bl\_array)

{

    int res;

    short k = 0;

    for(int i = s\_int-1; i >= 0; --i)

        if(new\_bl\_array[k++])   val |= (1<<i);

        else                    val &=~(1<<i);

    return val;

}

void task\_realize(bool \*bl\_array, bool \*res\_array, short s\_group, short index1, short index2, short amount)

{

/\*

    Реализация задания через последовательный перебор

    Сложность алгоритма O(n), где n - длина кодового слова

\*/

    int i;

    index1 = amount - index1 - 1;

    index2 = amount - index2 - 1;

    // Смотрим, чтобы выполнялось неравенство

    if(index1 > index2)

    {

        i = index2; index2 = index1; index1 = i;

    }

    i = 0;

    // Доходим до index1

    while(i != index1)

    {

        res\_array[i] = bl\_array[i];

        i++;

    }

    // Свап с index2

    while(i < index1+s\_group)

    {

        res\_array[i] = bl\_array[(index2) + (i) - index1];

        i++;

    }

    // Доходим до index2

    while(i != index2)

    {

        res\_array[i] = bl\_array[i];

        i++;

    }

    // Свап с index1

    while(i < index2+s\_group)

    {

        res\_array[i] = bl\_array[(index1) + (i) - index2];

        i++;

    }

    // Пробегаем по оставшимся

    while(i < amount)

    {

        res\_array[i] = bl\_array[i];

        i++;

    }

}

void print\_binary(bool \*bl\_array, short amount)

{

    int i;

    cout << "-----------------------\n";

    cout << "Binary code: ";

    for(i = 0; i <= amount-1; i++) cout << bl\_array[i];

    // if(amount != s\_int)

    // {

    //  cout << "\nsign: " << bl\_array[0];

    //  cout << "\nexponent: ";

    //  for (i = 1; i <= 15; i++) cout << bl\_array[i];

    //  cout << "\nmantissa: ";

    //  for (i = 16; i <= 79; i++) cout << bl\_array[i];

    // }

}

# 4. Примеры запуска программ

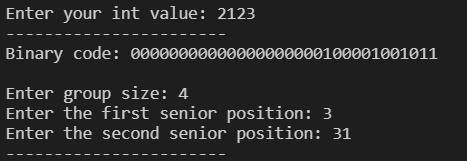


Рисунок 3 Пример запуска программы

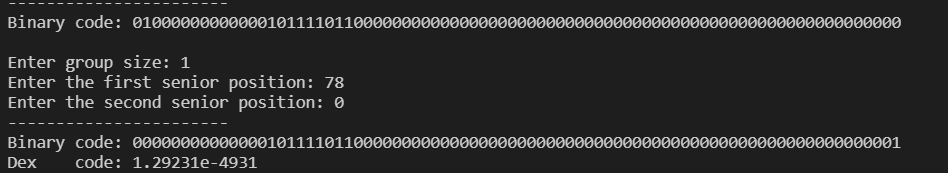


Рисунок 4 Пример запуска программы

# 5. Структурная схема аппаратных средств

