**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №1**

**по дисциплине «Организация ЭВМ и систем»**

**Тема: «Исследование внутреннего представления различных форматов данных»**

**Вариант 2**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3354 |  | Григорьев В.С. |
| Преподаватель |  | Анисимов А.В. |

Санкт-Петербург

2024

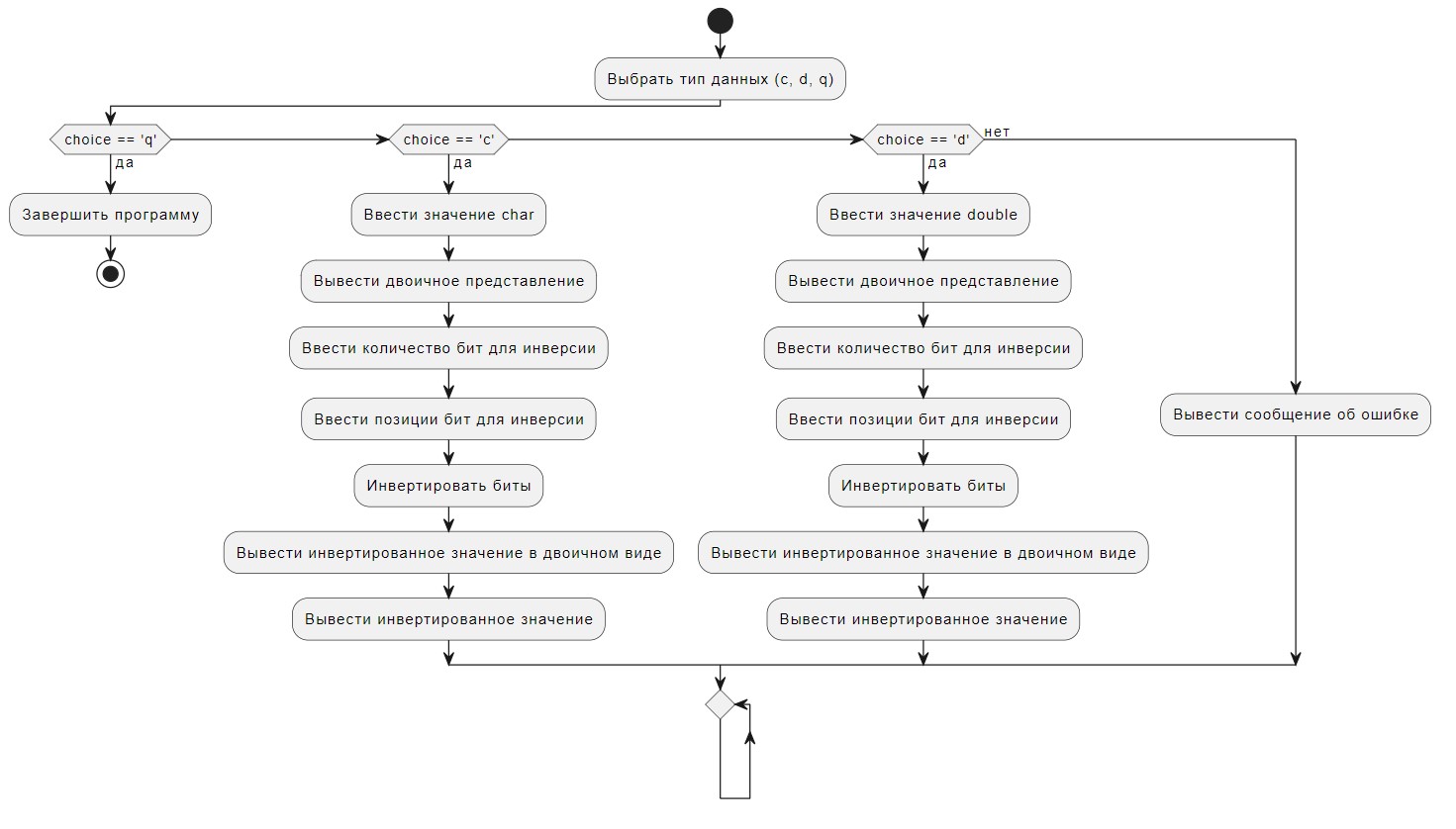
**Цель работы:** знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке.

**Задание:** разработать алгоритм ввода с клавиатуры требуемых типов данных и показать на экране их внутреннее представление в двоичной системе счисления. Написать и отладить программу на языке С++, реализующую разработанный алгоритм. 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.

**Типы данных:** char, double

**Вид преобразования:** Инвертировать значения определённого количества бит, номера которых, как и их количество, вводится с клавиатуры.

**Блок-схема**



**Текст программы**

#include <iostream>

#include <bitset>

#include <limits>

#include <vector>

#include <sstream>

#include <string>

using namespace std;

template<typename T>

void displayBinaryRepresentation(T value) {

const int bits = sizeof(T) \* 8;

bitset<bits> b(\*reinterpret\_cast<unsigned long long\*>(&value));

cout << "Двоичное представление: " << b << endl;

}

template<typename T>

T invertBits(T value, const vector<int>& positions) {

const int bits = sizeof(T) \* 8;

bitset<bits> b(\*reinterpret\_cast<unsigned long long\*>(&value));

for (int pos : positions) {

if (pos >= 0 && pos < bits) {

b.flip(pos);

}

}

return \*reinterpret\_cast<T\*>(&b);

}

template<typename T>

bool getInput(T& value) {

cin >> value;

if (cin.fail()) {

cin.clear(); // Сброс флага ошибки

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Очистка входного буфера

return false; // Ошибка ввода

}

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Очистка оставшегося ввода

return true; // Успешный ввод

}

int main() {

setlocale(LC\_ALL, "rus");

while (true) {

cout << "Выберите тип данных (c для char, d для double, q для выхода): ";

char choice;

cin >> choice;

if (choice == 'q' || choice == 'Q') {

break; // Выход из программы

}

if (choice == 'c' || choice == 'C') {

char value;

cout << "Введите значение типа char: ";

if (!getInput(value)) {

cout << "Ошибка: неверный ввод. Пожалуйста, введите один символ." << endl;

continue;

}

displayBinaryRepresentation(value);

const int max\_bits = sizeof(char) \* 8;

int num\_bits;

cout << "Введите количество бит для инверсии (максимум " << max\_bits << "): ";

if (!getInput(num\_bits) || num\_bits < 1 || num\_bits > max\_bits) {

cout << "Ошибка: количество бит должно быть от 1 до " << max\_bits << "." << endl;

continue;

}

vector<int> positions;

cout << "Введите позиции бит для инверсии (через запятую, например, 0,1,2): ";

string input;

getline(cin, input);

stringstream ss(input);

int pos;

while (ss >> pos) {

positions.push\_back(pos);

if (ss.peek() == ',') {

ss.ignore();

}

}

if (positions.size() != num\_bits) {

cout << "Ошибка: количество позиций для инверсии должно соответствовать введенному количеству бит." << endl;

continue;

}

char inverted\_value = invertBits(value, positions);

cout << "Инвертированное значение в двоичном виде: ";

displayBinaryRepresentation(inverted\_value);

cout << "Инвертированное значение: " << inverted\_value << endl;

}

else if (choice == 'd' || choice == 'D') {

double value;

cout << "Введите значение типа double: ";

if (!getInput(value)) {

cout << "Ошибка: неверный ввод. Пожалуйста, введите число." << endl;

continue;

}

displayBinaryRepresentation(value);

const int max\_bits = sizeof(double) \* 8;

int num\_bits;

cout << "Введите количество бит для инверсии (максимум " << max\_bits << "): ";

if (!getInput(num\_bits) || num\_bits < 1 || num\_bits > max\_bits) {

cout << "Ошибка: количество бит должно быть от 1 до " << max\_bits << "." << endl;

continue;

}

vector<int> positions;

cout << "Введите позиции бит для инверсии (через запятую, например, 0,1,2): ";

string input;

getline(cin, input);

stringstream ss(input);

int pos;

while (ss >> pos) {

positions.push\_back(pos);

if (ss.peek() == ',') {

ss.ignore();

}

}

if (positions.size() != num\_bits) {

cout << "Ошибка: количество позиций для инверсии должно соответствовать введенному количеству бит." << endl;

continue;

}

double inverted\_value = invertBits(value, positions);

cout << "Инвертированное значение в двоичном виде: ";

displayBinaryRepresentation(inverted\_value);

cout << "Инвертированное значение: " << inverted\_value << endl;

}

else {

cout << "Неверный выбор. Пожалуйста, выберите c для char, d для double или q для выхода." << endl;

}

cout << endl;

}

return 0;

}

**Примеры запуска программы**

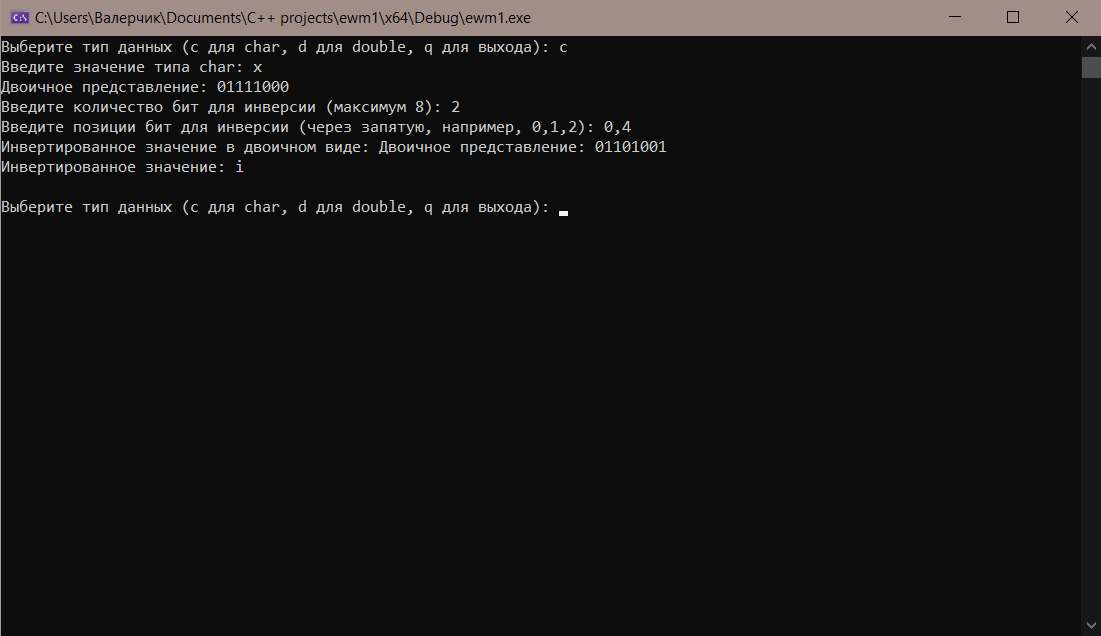
****

Рисунок 1. Пример запуска для типа char

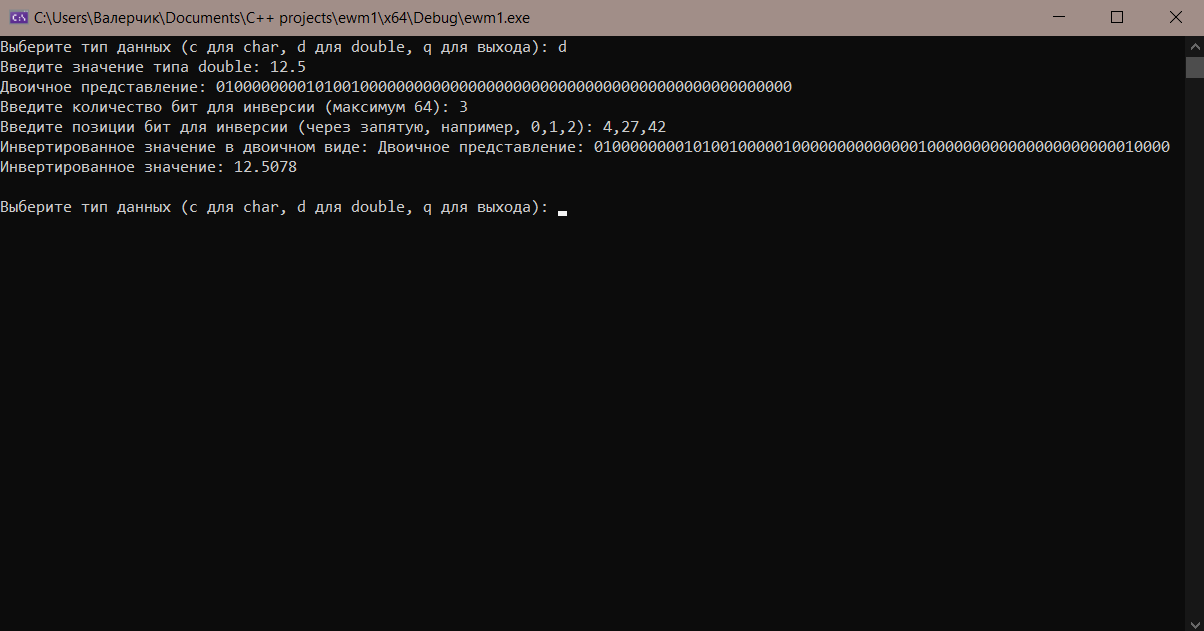
****

Рисунок 2. Пример запуска для типа double

**Программная среда**

Лабораторная работа выполнялась и компилировалась в программной среде Visual Studio 2022.