**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»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3315 |  | Купченко В.В. |
| Преподаватель |  | Анисимов А.В. |

Санкт-Петербург

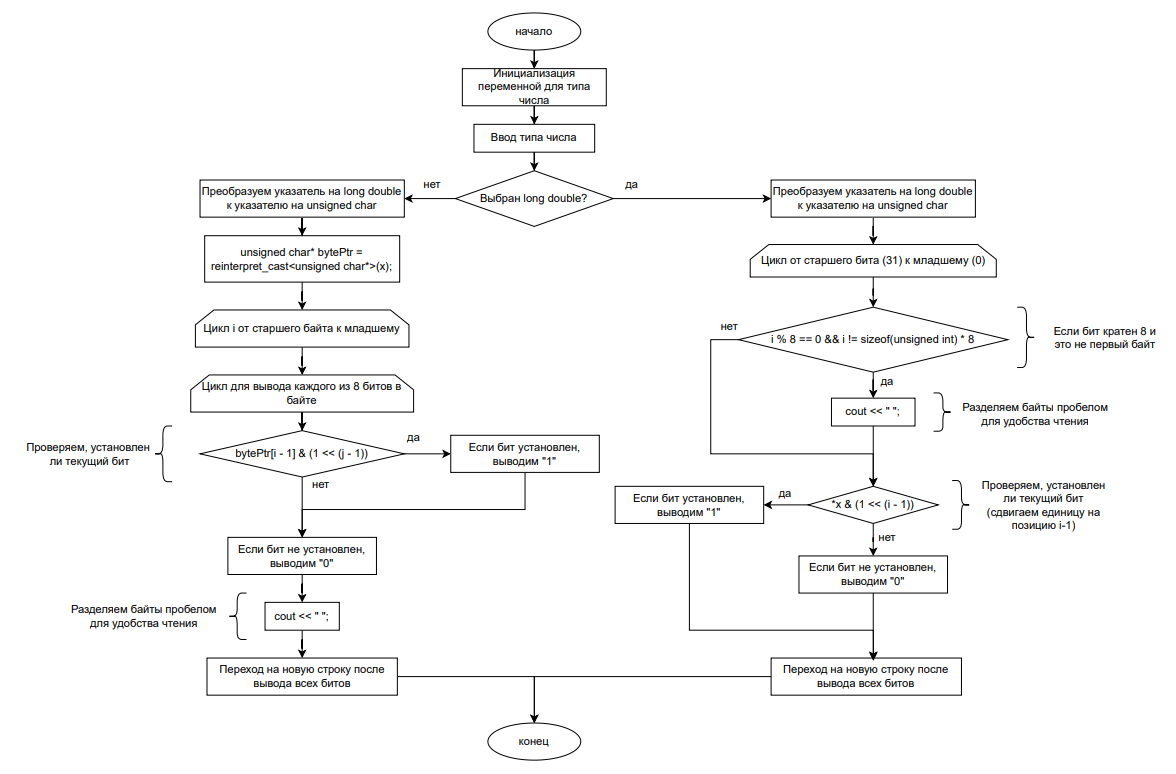
2024

**Задание (вариант 15)**

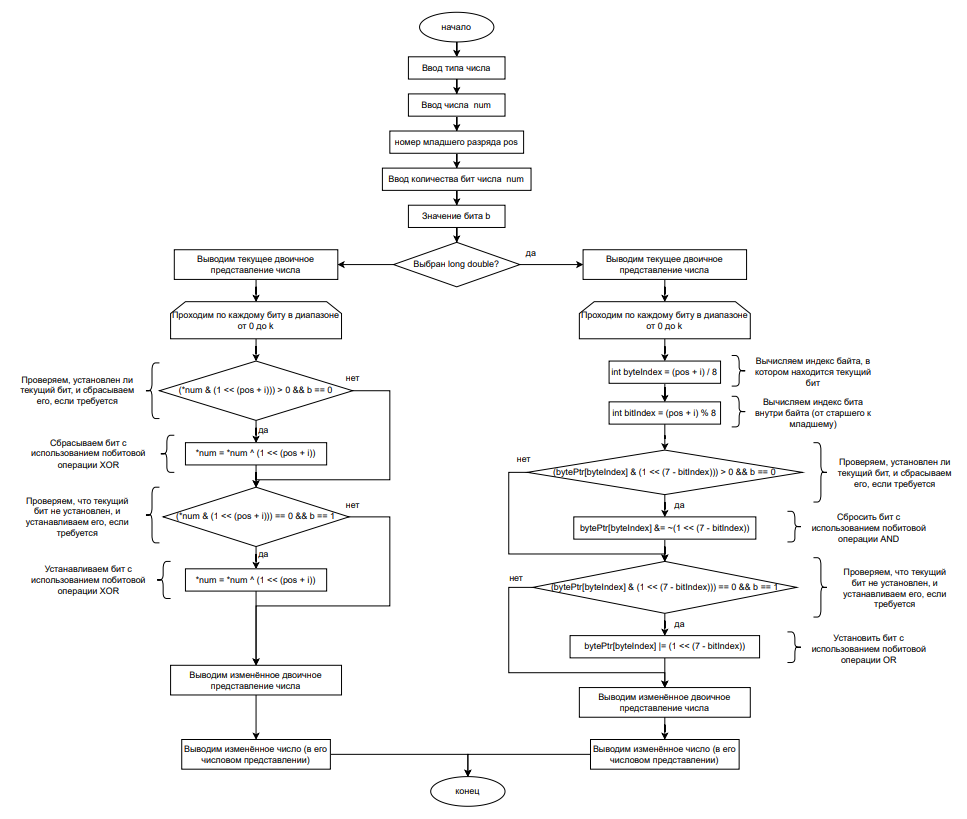
Разработать алгоритм ввода с клавиатуры данных типа long double, unsigned int и показать на экране их внутреннее представление в двоичной системе счисления. Написать и отладить программу на языке С++, реализующую разработанный алгоритм. В соответствии с заданием дополнить разработанный ранее алгоритм блоками для выполнения преобразования двоичного полученного кода исходного типа данных и последующего вывода преобразованного кода в двоичной системе счисления и в формате исходного данного.

**Блок-схема алгоритма**

Блок-схема для первого задания



Блок-схема для второго задания



**Текст программы**

#include<iostream>

#include<stdlib.h>

#include<vector>

using namespace std;

// Функция для вывода битового представления long double

void getBitRepresentation(long double\* x) {

// Преобразуем long double в массив байтов для вывода побитового представления

unsigned char\* bytePtr = reinterpret\_cast<unsigned char\*>(x);

for (size\_t i = sizeof(long double); i > 0; --i) {

for (size\_t j = 8; j > 0; --j) {

if (bytePtr[i - 1] & (1 << (j - 1))) {

cout << "1";

} else {

cout << "0";

}

}

cout << " ";

}

cout << endl;

}

// Функция для вывода битового представления unsigned int

void getBitRepresentation(unsigned int\* x) {

for (int i = sizeof(unsigned int) \* 8; i > 0; i--) {

if (i % 8 == 0 && i != sizeof(unsigned int) \* 8) {

cout << " ";

}

if (\*x & (1 << (i - 1))) {

cout << "1";

} else {

cout << "0";

}

}

cout << endl;

}

// Функция для изменения битов в long double

void changeBits(long double\* num, int pos, int k, int b) {

if (pos > 127 || pos < 0) { // Для long double 128 бит

cout << "Error: Invalid position" << endl;

return;

}

if (pos + k > 128) {

cout << "Error: Invalid range" << endl;

return;

}

// Выводим текущее двоичное представление числа

getBitRepresentation(num);

unsigned char\* bytePtr = reinterpret\_cast<unsigned char\*>(num);

for (size\_t i = 0; i < k; i++) {

int byteIndex = (pos + i) / 8;

int bitIndex = (pos + i) % 8;

if ((bytePtr[byteIndex] & (1 << (7 - bitIndex))) > 0 && b == 0) {

bytePtr[byteIndex] &= ~(1 << (7 - bitIndex)); // Сбросить бит

}

if ((bytePtr[byteIndex] & (1 << (7 - bitIndex))) == 0 && b == 1) {

bytePtr[byteIndex] |= (1 << (7 - bitIndex)); // Установить бит

}

}

// Выводим изменённое двоичное представление

getBitRepresentation(num);

cout << \*num << endl;

}

// Функция для изменения битов в unsigned int

void changeBits(unsigned int\* num, int pos, int k, int b) {

if (pos > 31 || pos < 0) { // Для unsigned int 32 бита

cout << "Error: Invalid position" << endl;

return;

}

if (pos + k > 32) {

cout << "Error: Invalid range" << endl;

return;

}

// Выводим текущее двоичное представление числа

getBitRepresentation(num);

for (size\_t i = 0; i < k; i++) {

if ((\*num & (1 << (pos + i))) > 0 && b == 0) {

\*num = \*num ^ (1 << (pos + i)); // Сбросить бит

}

if ((\*num & (1 << (pos + i))) == 0 && b == 1) {

\*num = \*num ^ (1 << (pos + i)); // Установить бит

}

}

// Выводим изменённое двоичное представление

getBitRepresentation(num);

cout << \*num << endl;

}

int main() {

cout << "Bit representation of what type would you like to get? Type 1 for long double, 2 for unsigned int" << endl;

int type;

cin >> type;

if (type == 1) {

long double d;

cout << "Enter number (long double):" << endl;

cin >> d;

getBitRepresentation(&d); // Выводим двоичное представление для long double

} else if (type == 2) {

unsigned int i;

cout << "Enter number (unsigned int):" << endl;

cin >> i;

getBitRepresentation(&i); // Выводим двоичное представление для unsigned int

} else {

cout << "Wrong type" << endl;

return 1;

}

cout << "Next task" << endl;

cout << "Enter number in which you would like to change bits. Type 1 for long double, 2 for unsigned int" << endl;

cin >> type;

if (type == 1) {

long double d;

cout << "Enter number (long double):" << endl;

cin >> d;

cout << "Enter position of 1st bit to change (numbering starts from right):" << endl;

int pos;

cin >> pos;

int num;

cout << "Enter how many bits do you want to change?" << endl;

cin >> num;

int b;

cout << "Enter the new value for the bits (0 or 1):" << endl;

cin >> b;

changeBits(&d, pos, num, b);

} else if (type == 2) {

unsigned int i;

cout << "Enter number (unsigned int):" << endl;

cin >> i;

cout << "Enter position of 1st bit to change (numbering starts from right):" << endl;

int pos;

cin >> pos;

int num;

cout << "Enter how many bits do you want to change?" << endl;

cin >> num;

int b;

cout << "Enter the new value for the bits (0 or 1):" << endl;

cin >> b;

changeBits(&i, pos, num, b);

} else {

cout << "Error, wrong type" << endl;

return 1;

}

return 0;

}

**Пример работы программы**

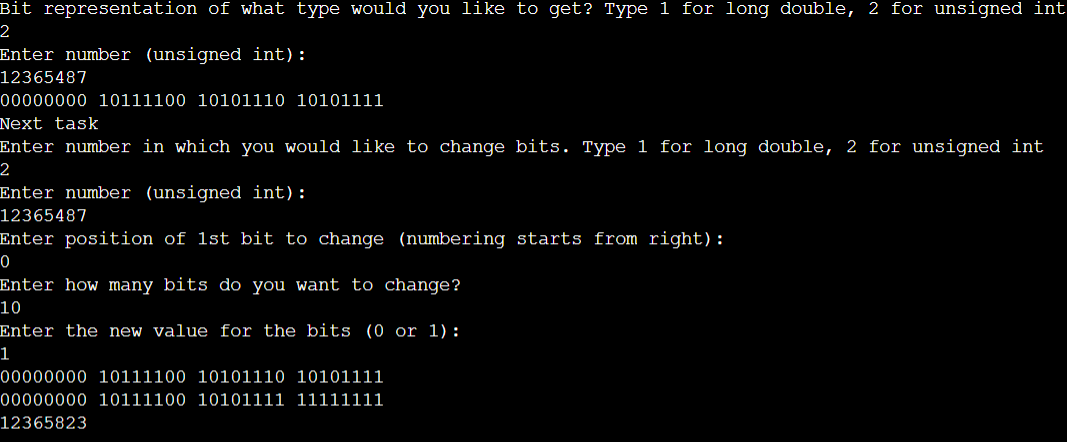
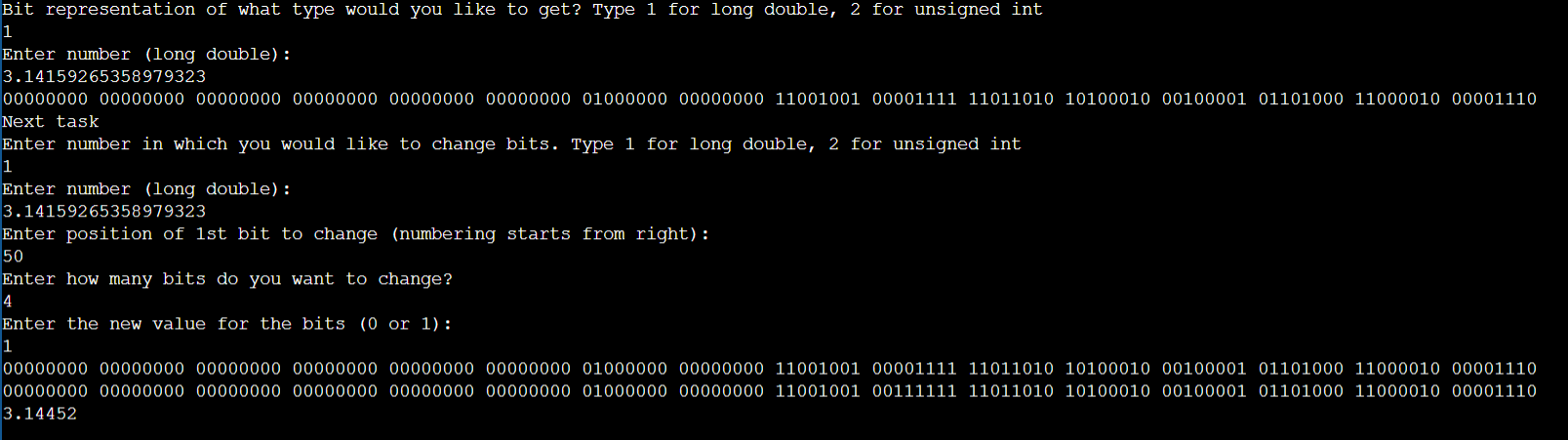
**** Рисунок 1. Тип данных: unsigned int

Рисунок 2. Тип данных: long double