**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: И**сследование внутреннего представления различных форматов данных**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3353 |  | Карпенко А.Ю. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

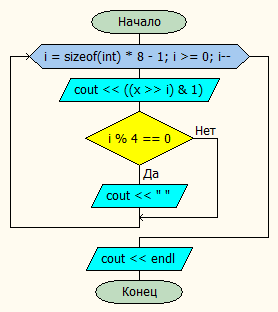
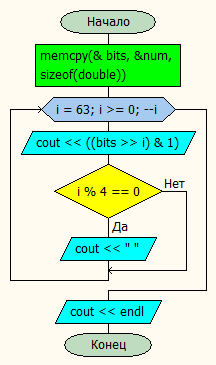
2024

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

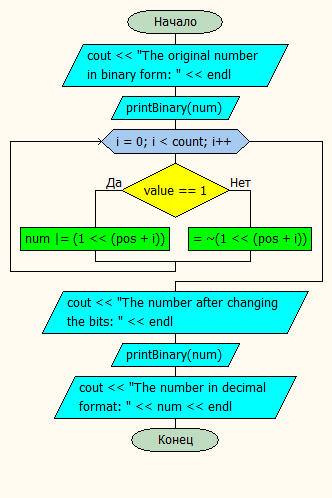
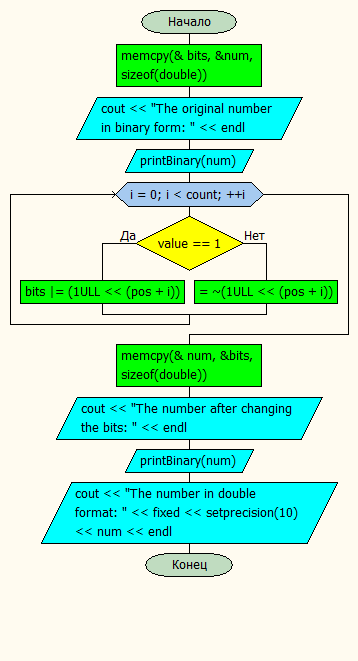
**Задание:** разработать алгоритм ввода с клавиатуры требуемых типов данных и показать на экране их внутреннее представление в двоичной системе счисления. Написать и отладить программу на языке С++, реализующую разработанный алгоритм. В соответствии с заданием дополнить разработанный ранее алгоритм блоками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

**Блок-схема алгоритма**

1. Блок-схема для 1 задания

1. Блок-схема для 2 задания

**Текст программы**

#include <iostream>

#include <cstdint>

using namespace std;

// Функция для вывода двоичного представления int

void printBinary(int x) {

for (int i = sizeof(int) \* 8 - 1; i >= 0; i--) {

cout << ((x >> i) & 1);

/\*

" - Операция `x >> i` сдвигает число `x` вправо на `i` битов. Это позволяет нам на каждой итерации получать бит, находящийся в позиции `i`.\n"

" - Операция `& 1` используется для того, чтобы оставить только самый младший бит (0 или 1), так как побитовая операция \"и\" с числом 1 отсекает все остальные биты.\n"

" - Таким образом, на каждой итерации цикла выводится 1 или 0, в зависимости от значения бита в позиции `i`."\*/

if (i % 4 == 0) { // Для удобства чтения разделили по 4 бита

cout << " ";

}

}

cout << endl;

}

// Функция для изменения битов в числе

/\*

int& num — это ссылка на целое число, биты которого нужно изменить. Использование ссылки (&) позволяет изменять исходное число в вызывающем коде.

int pos — позиция, начиная с которой будет изменяться биты числа. Позиции считаются от младших к старшим битам, начиная с 0.

int count — количество битов, которые нужно изменить.

int value — значение, на которое будут установлены биты (0 или 1).\*/

void changeBits(int& num, int pos, int count, int value) {

cout << "The original number in binary form: " << endl;

printBinary(num);

for (int i = 0; i < count; i++) {

if (value == 1) {

num |= (1 << (pos + i)); // Set bit to 1

/\*|= ... — операция побитового "ИЛИ" с числом num. Эта операция устанавливает в 1 бит в позиции,

\* заданной маской, сохраняя все остальные биты без изменений.\*/

} else {

num &= ~(1 << (pos + i)); // Set bit to 0

/\* &= ... — операция побитового "И" с числом num и инверсированной маской.

\* Эта операция сбрасывает бит в позиции (pos + i) в 0, а все остальные биты остаются неизменными.\*/

}

}

cout << "The number after changing the bits: " << endl; //Число после изменения битов

printBinary(num);

cout << "The number in decimal format: " << num << endl; //Число в десятичном формате

}

// Функция для вывода двоичного представления double

void printBinary(double num) {

uint64\_t bits;

memcpy(&bits, &num, sizeof(double));

/\*uint64\_t bits — переменная типа uint64\_t, которая представляет 64-битное целое число без знака.

\* Мы используем это, так как числа типа double занимают 64 бита, и нам нужно с ними работать на уровне битов.

memcpy(&bits, &num, sizeof(double)) — функция memcpy копирует байты из одного места памяти в другое.

В данном случае мы копируем байты числа num в переменную bits.

Это нужно, потому что напрямую работать с битами числа с плавающей точкой нельзя,

но можно их интерпретировать как целое число и работать с битами напрямую.\*/

for (int i = 63; i >= 0; --i) {

cout << ((bits >> i) & 1);

if (i % 4 == 0) {

cout << " ";

}

}

cout << endl;

}

// Функция для изменения битов в double

void changeBits(double& num, int pos, int count, int value) {

uint64\_t bits;

memcpy(&bits, &num, sizeof(double));

cout << "The original number in binary form: " << endl;

printBinary(num);

// Изменение битов

for (int i = 0; i < count; ++i) {

if (value == 1) {

bits |= (1ULL << (pos + i));

/\*Если value == 1, то используется операция побитового ИЛИ (|=), которая устанавливает бит на нужной позиции в 1.

\* Операция сдвига 1ULL << (pos + i) создает число, где только бит на позиции pos + i равен 1, а все остальные — 0.

\* ИЛИ с этим числом устанавливает бит в 1, оставляя все остальные биты без изменений.\*/

} else {

bits &= ~(1ULL << (pos + i));

/\*Если value == 0, то используется операция побитового И (&=) с инверсией маски ~(1ULL << (pos + i)),

\* которая сбрасывает бит на нужной позиции в 0, не затрагивая остальные.\*/

}

}

memcpy(&num, &bits, sizeof(double));

cout << "The number after changing the bits: " << endl;

printBinary(num);

cout << "The number in double format: " << num << endl;

}

int main() {

cout << "Select the number type: 1-int, 2-double" << endl; //выберете тип числа: 1-int, 2-double

int k;

cin >> k;

if (k == 1) {

int num;

cout << "Enter an integer: "; //Введите целое число:

cin >> num;

cout << "Binary representation of the number: \n"; //Двоичное представление числа:

printBinary(num);

int pos, count, value;

cout << "Enter the position of the first bit to change (starting from 0): "; //Введите позицию первого бита для изменения (нумерация с 0):

cin >> pos;

cout << "Enter the number of bits to change: "; //Введите количество битов для изменения:

cin >> count;

cout << "Enter the value to set (0 or 1): "; //Введите значение для установки (0 или 1):

cin >> value;

changeBits(num, pos, count, value);

} else if (k == 2) {

double num;

cout << "Enter a number of type double: "; //Введите число типа double:

cin >> num;

cout << "Binary representation of the number: \n"; //Двоичное представление числа:

printBinary(num);

int pos, count, value;

cout << "Enter the position of the first bit to change (starting from 0): ";//Введите позицию первого бита для изменения (нумерация с 0):

cin >> pos;

cout << "Enter the number of bits to change: ";//Введите количество битов для изменения:

cin >> count;

cout << "Enter the value to set (0 or 1): ";//Введите значение для установки (0 или 1):

cin >> value;

changeBits(num, pos, count, value);

} else {

cout << "Invalid input" << endl; //неверно введенное число

}

return 0;

}

**Примеры запуска программы**

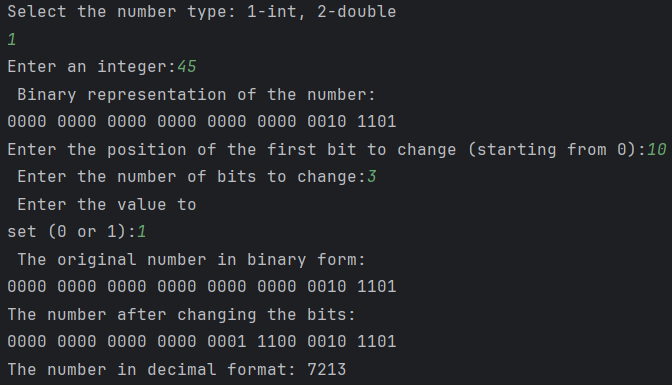


Рисунок 1 – Тип int

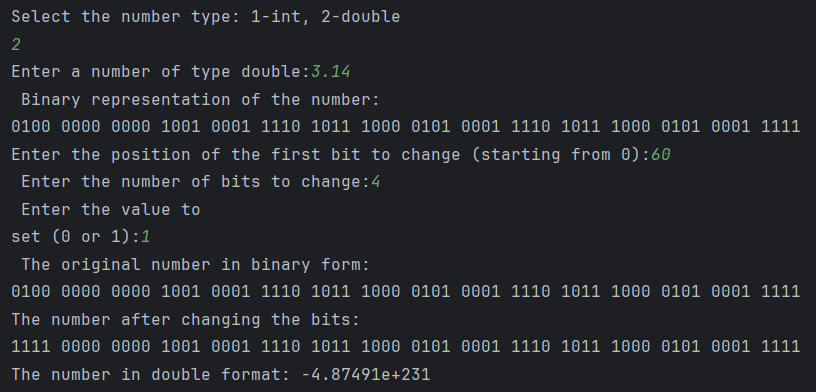


Рисунок 2 – Тип double

**Структурная схема аппаратных средств, используемых при выполнении программы с необходимой степенью детализации содержимого блоков.**

ОЗУ

Устройство  
ввода

ЦП

Устройство  
вывода

Контроллер ввода

Контроллер вывода

Системная шина

**Центральный процессор (ЦП)**:

Центральный процессор (или процессор) выполняет инструкции программы. ЦП обменивается данными с оперативной памятью, управляющими устройствами и устройствами ввода/вывода.

**Оперативная память (ОЗУ)**:

Оперативная память временно хранит данные и инструкции, которые используются процессором во время выполнения программы. Процессор извлекает инструкции и данные из памяти для их обработки.

**Системная шина**:

Системная шина обеспечивает обмен данными между процессором, памятью и устройствами ввода/вывода. Соединяет все компоненты компьютера для передачи данных и команд.

**Устройства ввода**:

К устройствам ввода относятся клавиатура, мышь, сканеры и другие устройства, с помощью которых пользователь вводит данные в систему. Данные с устройств ввода поступают в процессор через контроллер ввода/вывода.

**Устройства вывода**:

Устройства вывода, такие как мониторы, принтеры и колонки, служат для вывода обработанной информации пользователю. Получают данные от процессора через контроллер вывода.

**Контроллеры ввода/вывода**:

Эти устройства управляют обменом данными между процессором и периферийными устройствами. Подключены к шине и обеспечивают связь с внешними устройствами.