**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. 1376 |  | Прохоренко К.Д. |
| Преподаватель |  | Ельчанинов М.Н. |

Санкт-Петербург

2022

**Цель работы.**

Знакомство с внутренним представлением различных типов данных, используемых компьютером при их обработке.

**Задание на лабораторную работу.**

1. Разработать алгоритм ввода с клавиатуры типов данных **short** и **long double** 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.

Вид преобразования: Поменять местами значения бит в заданном количестве пар бит. Номера бит в парах задаются с клавиатуры.

**Текст программы**

#include <iostream>

#include <conio.h>

#include <string>

#include <vector>

**using** **namespace** std;

**union** longdouble

{

long double ld;

\_\_int128\_t ldint;

};

void tobinary(char data)

{

unsigned char mask;

**for** (mask = 128; mask != 0; mask >>= 1)

{

bool a = (data & mask);

std::cout << a;

}

std::cout << " ";

}

void output(short data)

{

**for** (short i = **sizeof**(data); i != 0; i--)

{

tobinary(data >> (8 \* (i - 1)));

}

}

void output(\_\_int128\_t data)

{

**for** (short i = **sizeof**(data) - 6; i != 0; i--)

{

tobinary(data >> (8 \* (i - 1)));

}

}

bool range(char s)

{

**if** ((s >= '0' && s <= '9') ) **return** 1;

**else** **return** 0;

}

long double input(long double data)

{

char s;

string result;

**do**

{

s = \_getch();

**if** (range(s))

{

std::cout << char(s);

result.push\_back(s);

}

**else** **if** (s == 8)

{

**if** (!result.empty())

{

result.pop\_back();

std::cout << "**\b** **\b**";

}

}

**else** **if** (s == '-' && result.empty())

{

std::cout << char(s);

result.push\_back(s);

}

**else** **if** (s == '.')

{

std::cout << char(s);

result.push\_back(s);

}

} **while** (s != 13);

std::cout << endl;

long double ans = 0;

long double ans\_f = 0;

short sign = 1;

**if** (result[0] == '-')

{

sign = -1;

result.erase(0,1);

}

**for**(int i = 0; i < result.size(); i++)

{

**if** (result[i] == 0) result.erase(0,1);

**else** **break**;

}

**for**(int i = 0; i < result.size(); i++)

{

**if**(result[i] == '.') **break**;

ans = ans \* 10 + result[i] - '0';

}

**while** (result.back() != '.')

{

ans\_f = (ans\_f + (result.back())- '0') / 10;

result.pop\_back();

}

ans = sign \* (ans + ans\_f);

**return** ans;

}

short input(short data)

{

char s;

string result;

**do**

{

s = \_getch();

**if** (range(s))

{

std::cout << char(s);

result.push\_back(s);

}

**else** **if** (s == 8)

{

**if** (!result.empty())

{

result.pop\_back();

std::cout << "**\b** **\b**";

}

}

**else** **if** (s == '-' && result.empty())

{

std::cout << char(s);

result.push\_back(s);

}

} **while** (s != 13);

std::cout << endl;

short ans = 0;

short sign = 1;

**if** (result[0] == '-')

{

sign = -1;

result.erase(0,1);

}

**for**(int i = 0; i < result.size(); i++)

{

**if** (result[i] == 0) result.erase(0,1);

**else** **break**;

}

**for**(int i = 0; i < result.size(); i++)

{

ans = ans \* 10 + result[i] - '0';

}

**return** ans\*sign;

}

int swap\_bits(int data, int i, int j)

{

bool biti = data & (1 << i);

bool bitj = data & (1 << j);

**if** (biti != bitj)

{

int bit\_mask = (1 << i) | (1 << j);

data ^= bit\_mask;

**return** data;

}

**return** data;

}

longdouble swap\_bits(longdouble data, int i, int j)

{

bool biti = data.ldint & ((\_\_int128\_t)1 << i);

bool bitj = data.ldint & ((\_\_int128\_t)1 << j);

**if** (biti != bitj)

{

\_\_int128\_t bit\_mask = ((\_\_int128\_t)1 << i) | ((\_\_int128\_t)1 << j);

data.ldint ^= bit\_mask;

**return** data;

}

**return** data;

}

short reversepairs(short data)

{

char s;

string result;

**do**

{

s = \_getch();

**if** (s >= '0' && s <= '9')

{

std::cout << s;

result.push\_back(s);

}

**else** **if** (s == 8)

{

**if** (!result.empty())

{

result.pop\_back();

std::cout << "**\b** **\b**";

}

}

**else** **if** (s == 32)

{

**if**(!result.empty())

{

result.push\_back(s);

std::cout << " ";

}

}

} **while** (s != 13);

vector <int> vresult;

int value = 0;

**for**(int i = 0; i < result.size(); i++)

{

**if**(result[i] != char(32))

{

value = value \* 10 + (result[i] - '0');

}

**if**(result[i] == char(32) || i == result.size() - 1)

{

vresult.push\_back(value);

value = 0;

}

}

**for**(int i = 0; i < result.size(); i+=2)

{

data = swap\_bits(data, vresult[i], vresult[i+1]);

}

**return** data;

}

longdouble reversepairs(longdouble data)

{

char s;

string result;

**do**

{

s = \_getch();

**if** (s >= '0' && s <= '9')

{

std::cout << s;

result.push\_back(s);

}

**else** **if** (s == 8)

{

**if** (!result.empty())

{

result.pop\_back();

std::cout << "**\b** **\b**";

}

}

**else** **if** (s == 32)

{

**if**(!result.empty())

{

result.push\_back(s);

std::cout << " ";

}

}

} **while** (s != 13);

vector <int> vresult;

int value = 0;

**for**(int i = 0; i < result.size(); i++)

{

**if**(result[i] != char(32))

{

value = value \* 10 + (result[i] - '0');

}

**if**(result[i] == char(32) || i == result.size() - 1)

{

vresult.push\_back(value);

value = 0;

}

}

**for**(int i = 0; i < vresult.size(); i += 2)

{

data = swap\_bits(data, vresult[i], vresult[i+1]);

}

**return** data;

}

int main()

{

int menu;

**do**

{

*system("cls");*

std::cout << "1. short**\n**";

std::cout << "2. long double**\n**";

std::cout << "0. exit**\n**";

menu = \_getch();

**if**(menu == '1')

{

short data = 0;

std::cout << "input short int number: ";

data = input(data);

std::cout << "decimal: " << data << std::endl;

std::cout << "binary: ";

output(data);

std::cout << std::endl;

std::cout << "enter the number of bit pairs to be changed: ";

short reversedBits = reversepairs(data);

std::cout << "**\n\n**decimal: " << reversedBits << std::endl;

std::cout << "binary: ";

output(reversedBits); std::cout << std::endl << std::endl;

}

**if**(menu == '2')

{

longdouble data;

std::cout << "**\n**input float number: ";

data.ld = input(data.ld);

std::cout << "decimal: " << data.ld << endl;

std::cout << "binary: ";

output(data.ldint);

std::cout << endl;

std::cout << "enter the number of bit pairs to be changed: ";

longdouble reversedBits = reversepairs(data);

std::cout << "**\n\n**decimal: " << reversedBits.ld << std::endl;

std::cout << "binary: ";

output(reversedBits.ldint); std::cout << std::endl << std::endl;

}

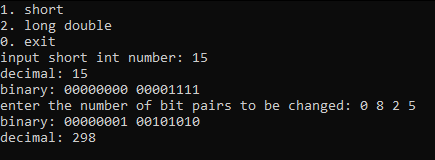
}

**while** (menu != '0');

}

**Примеры запуска программы**

Пример работы с short числом:

****

Пример работы с long double числом:

