**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. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2019

**Цель работы.**

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

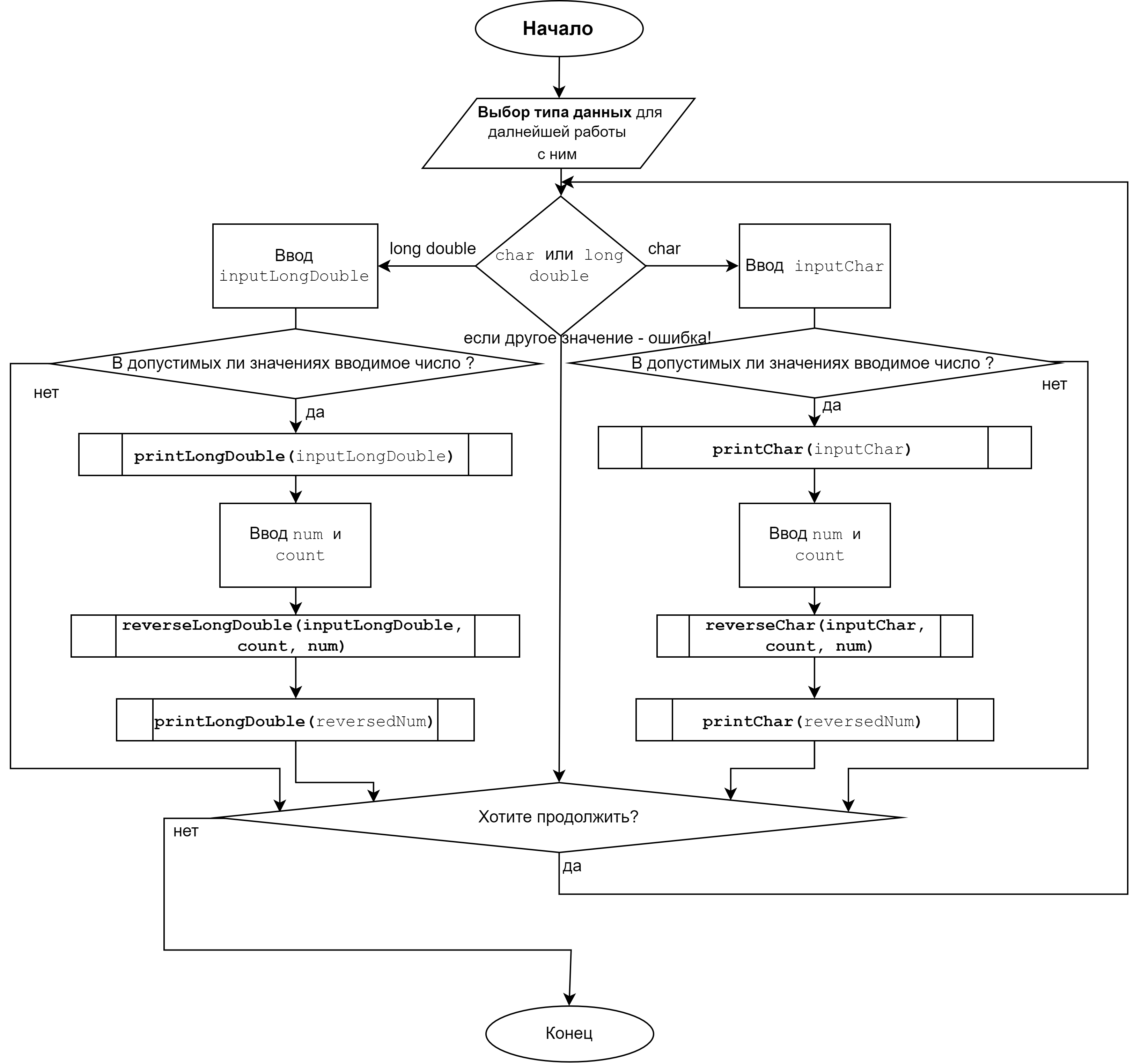
**Задание на лабораторную работу.**

1. Разработать алгоритм ввода с клавиатуры типов данных **char** 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: Выполнить зеркальную перестановку в группе рядом стоящих разрядов, количество которых и номер старшего разряда в группе задаются с клавиатуры.

**Блок-схема алгоритма**

**Текст программы**

#include <iostream>

using namespace std;

union type

{

unsigned long long ll;

long double d;

};

void output(unsigned long long x)

{

int i;

for (i = 8 \* sizeof(x) - 1; i > -1; i--)

printf("%d", x >> i & 1);

printf("\n");

}

void outputon(unsigned long long x)

{

int i;

for (i = 0; i > -1; i--)

printf("%d", x >> i & 1);

printf("\n");

}

void outputtw(unsigned long long x)

{

int i;

for (i = 11 - 1; i > -1; i--)

printf("%d", x >> i & 1);

printf("\n");

}

void outputtr(unsigned long long x)

{

int i;

for (i = 52 - 1; i > -1; i--)

printf("%d", x >> i & 1);

printf("\n");

}

typedef union {

long double ld;

struct {

unsigned long long mantisa : 52;

unsigned long long exponent : 11;

unsigned long long sign : 1;

} parts;

} ld\_cast;

void print(ld\_cast d1) {

printf("sign = ");

outputon(d1.parts.sign);

printf("exponent = ");

outputtw(d1.parts.exponent);

printf("mantisa = ");

outputtr(d1.parts.mantisa);

}

void printLongDouble(long double x)

{

cout << endl << endl;

{

type m;

m.d = x;

for (int i = 8 \* sizeof(m.ll) - 1; i > -1; i--)

printf("%d", m.ll >> i & 1);

printf("\n");

}

}

void printChar(char input)

{

cout << endl << endl;

unsigned int bit;

for (int i = 0; i < 8; i++)

{

bit = ((input >> (8 - i - 1)) & 1);

cout << bit;

}

cout << endl;

}

/\*

Что творится в функции: reverseLongDouble

Пусть count = 4, num = 6, возьмем для примера 8 бит, обозначим биты буквами.

ABCDEFGH

т.е. нам нужно развернуть BCDE, не трогая остальное

shift = 3 (длина хвоста FGH)

mask = 00001111 << 3 = 01111000 //маска для нужных битов

ext = (ABCDEFGH & 01111000) >> 3 = 0000BCDE //выделили, сдвинули вправо, чтобы ничего не мешало

revext = 0000EDCB //после цикла биты занесены в обратном порядке

rev\_ext <<= shift = 0EDCB000 // вернули обратно на место

ABCDEFGH & !mask = A0000FGH //сбросили нужную часть

A0000FGH | 0EDCB000 = AEDCBFGH //и вставили развернутую

\*/

unsigned long long reverseLongDouble(long double inputLongDouble, int count, int num)

{

unsigned long long shift = (unsigned long long)num - (unsigned long long)count + 1; // Visual Studio: мол Warning C26451

type inLongDouble;

inLongDouble.d = inputLongDouble;

unsigned long long mask = ((1 << (unsigned long long)count) - 1) << shift;

unsigned long long ext = (inLongDouble.ll & mask) >> shift;

unsigned long long rev\_ext = 0;

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

{

rev\_ext <<= 1;

rev\_ext |= ext & 1;

ext >>= 1;

}

rev\_ext <<= shift;

return (inLongDouble.ll & ~mask) | rev\_ext;

}

unsigned int reverseChar(unsigned int inputChar, int c, int n)

{

int shift = n - c + 1;

unsigned int mask = ((1 << c) - 1) << shift;

unsigned int ext = (inputChar & mask) >> shift;

unsigned int rev\_ext = 0;

int i;

for (i = 0; i < c; ++i) {

rev\_ext <<= 1;

rev\_ext |= ext & 1;

ext >>= 1;

}

rev\_ext <<= shift;

return (inputChar & ~mask) | rev\_ext;

}

int main()

{

setlocale(LC\_ALL, "rus");

char inputChar;

long double inputLongDouble;

int m, t;

do

{

t = 0;

cout << "Выберите с каким типом вы хотите работать: 1 - long double; 2 - char " << endl;

cin >> m;

if (m == 1)

{

cout << "Введите данные:\n";

cin >> inputLongDouble;

if ((inputLongDouble > -9223372036854775808.0) & (inputLongDouble < 9223372036854775807.0))

{

printLongDouble(inputLongDouble);

ld\_cast input;

input.ld = inputLongDouble;

print(input);

int count;

cout << "\nВведите количество для перестановки в группе рядом стоящих разрядов:";

cin >> count;

int num;

cout << "\nВведите номер старшего разряда:";

cin >> num;

output(reverseLongDouble(inputLongDouble, count, num));

}

else

{

cout << "\nОшибка!!!\n";

}

}

else if (m == 2)

{

cout << "Введите данные:\n";

cin >> inputChar;

if ((inputChar > -128) & (inputChar < 127))

{

printChar(inputChar);

int count;

cout << "Введите разрядов количество для перестановки в группе рядом стоящих разрядов: ";

scanf\_s("%d", &count);

int num;

cout << "Введите номер старшего разряда: ";

scanf\_s("%d", &num);

char reversed = reverseChar(inputChar, count, num);

printf("\n");

printf("%u\n", reversed);

printChar(reversed);

}

else

{

cout << "\nОшибка!!!\n";

}

}

else

{

cout << "\nОшибка!!!\n";

}

cout << "Хотите продолжить? да - 1; нет - 0\n";

cin >> t;

system("cls");

}

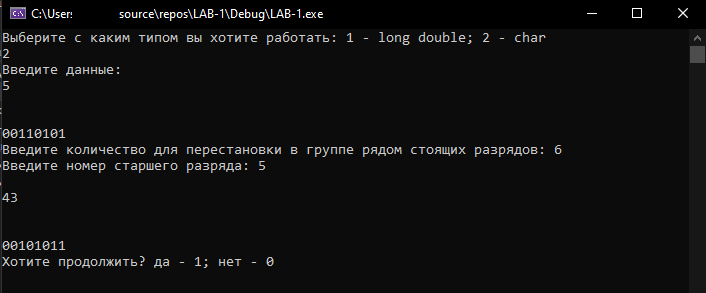
while (t != 0);

return 0;

}

**Примеры запуска программы**

На рисунке демонстрирована работа программы, сначала нужно выбрать тип данных с которым будут выполняться дальнейшие операции. Как показанно на рисунке после выбора типа пользователя предлагается вводить данные, после чего программа выводит на экран внутреннее представление этих данных в двоичной системе счисления. Пользователью также предлагаеться ввести количество разрядов для перестановки в группе рядом стоящих разрядо и номер старшего разряда. При получении необходимых данных программа выводит на экран выполненную зеркальную перестановку в группе рядом стоящих разрядов, количество которых и номер старшего разряда в группе было задано с клавиатуры.



Стоит отметить, что в программе учтены случаи ошибки, которые могли возникнуть в процессе эксплутации программы (неверно введенные данные и/или при ограниченном количестве действиий, например: выбор типа данных char, long double: при вводе данных отличных от ''1'' или ''2'' программа выдаст ошибку и выведет сообщение: ''Хотите продолжить?'', при ответе ''1'' программа начнет работу сначала, при ''0'' – завершит работу.

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

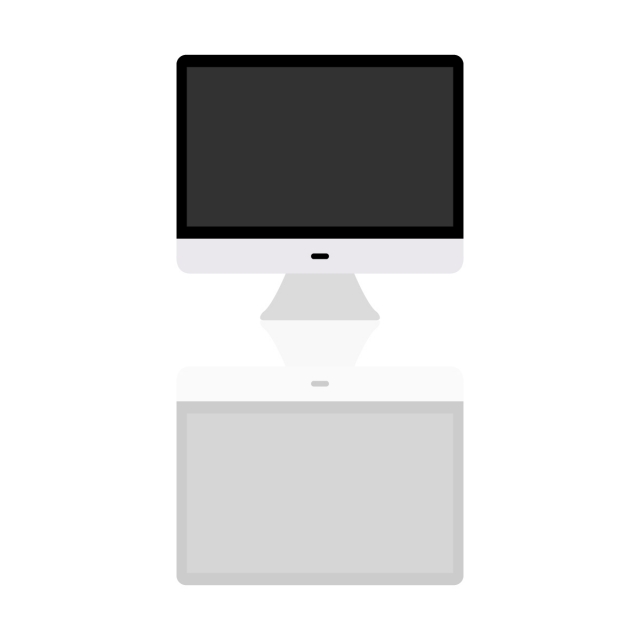
**ПЗУ**

**(RОM)**

**ОЗУ**

**(RAM)**

Контроллер

**ИП**

Клавиатура

Контроллер

Контроллер

**ЦП**

**(CPU)**

СИСТЕМНАЯ ШИНА