**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»**

Тема: **ИССЛЕДОВАНИЕ ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ РАЗЛИЧНЫХ ФОРМАТОВ ДАННЫХ**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0302 |  | Савенко Н.С |
| Преподаватель |  | Жандаров В. В. |

Санкт-Петербург

2021

**Цель работы.**

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

**Задание (вариант XV)**

Выполнить циклический сдвиг в заданную пользователем сторону на некоторое количество разрядов в пределах определённой группы разрядов, количество которых и номер младшего разряда в группе задаются с клавиатуры.

**Постановка задачи и описание решения**

Чередованием считаются последовательно идущие элементы, где любые два соседних имеют разный остаток при делении на 2.

Производится ввода массива.

Массив построчно проходит через на чередование элементов в строке. Если чередование нарушено, то строка не добавляется в новый массив, иначе строка переворачивается и добавляется в массив.

Производится вывод массива.

**Текст программы**

#include <iostream>;

#include <iomanip>;

#include <bitset>;

using namespace std;

template <size\_t bitsetsize>

bool shift\_bitset(bitset<bitsetsize>\* source\_bitset);

int main()

{

int number\_type;

cout << "Whith type do you use?(1 - unsigned int; 0 - long double):" << endl;

cin >> number\_type;

unsigned int \_uint;

long double \_ldouble;

cout << "Input source\n";

if (number\_type == 1) {

cin >> \_uint;

cout << "Source(\_uint): " << \_uint << endl;

auto bits = bitset<sizeof(unsigned int) \* 8>(\_uint);

cout << "bin(\_uint): " << bits << endl;

if (shift\_bitset<sizeof(unsigned int) \* 8>(&bits)) {

cout << "New value:\n" << (unsigned int)(bits.to\_ulong());

}

}

else if (number\_type == 0) {

cin >> \_ldouble;

cout << "Source(long double): " << \_ldouble << endl;

unsigned long long bits\_source = \*reinterpret\_cast<unsigned long long\*>(&\_ldouble);

auto bits = bitset<sizeof(long double) \* 8>(bits\_source);

cout << "bin(\_ldoube): " << bits << endl;

if (shift\_bitset<sizeof(long double) \* 8>(&bits)) {

cout << "New value:\n" << (long double)(bits.to\_ullong());

}

}

else {

cout << "Uncorrect type!" << endl;

}

return 0;

}

template <size\_t bitsetsize>

bool shift\_bitset(bitset<bitsetsize>\* source\_bitset) {

cout << "Enter start" << "\n";

int start;

cin >> start;

if (start < 0 || start >= bitsetsize) {

cout << "Invalid start position" << endl;

return false;

}

cout << "Enter area lenght" << "\n";

int len;

cin >> len;

if (len < 0 || len + start >= bitsetsize) {

cout << "Invalid area len" << endl;

return false;

}

cout << "Enter shift count(>0 move right; <0 move left)" << "\n";

int l\_count;

cin >> l\_count;

if (l\_count == 0) {

cout << "No shift" << "\n";

return false;

}

if (l\_count < 0) {

l\_count \*= -1;

for (auto i = 0; i < l\_count; i++) {

bool start\_bit = (\*source\_bitset)[start + len];

for (auto j = start + len; j > start; j--) {

(\*source\_bitset)[j] = (\*source\_bitset)[j - 1];

}

(\*source\_bitset)[start] = start\_bit;

}

}

else {

for (auto i = 0; i < l\_count; i++) {

bool start\_bit = (\*source\_bitset)[start];

for (auto j = start; j < start + len - 1; j++) {

(\*source\_bitset)[j] = (\*source\_bitset)[j + 1];

}

(\*source\_bitset)[start + len - 1] = start\_bit;

}

}

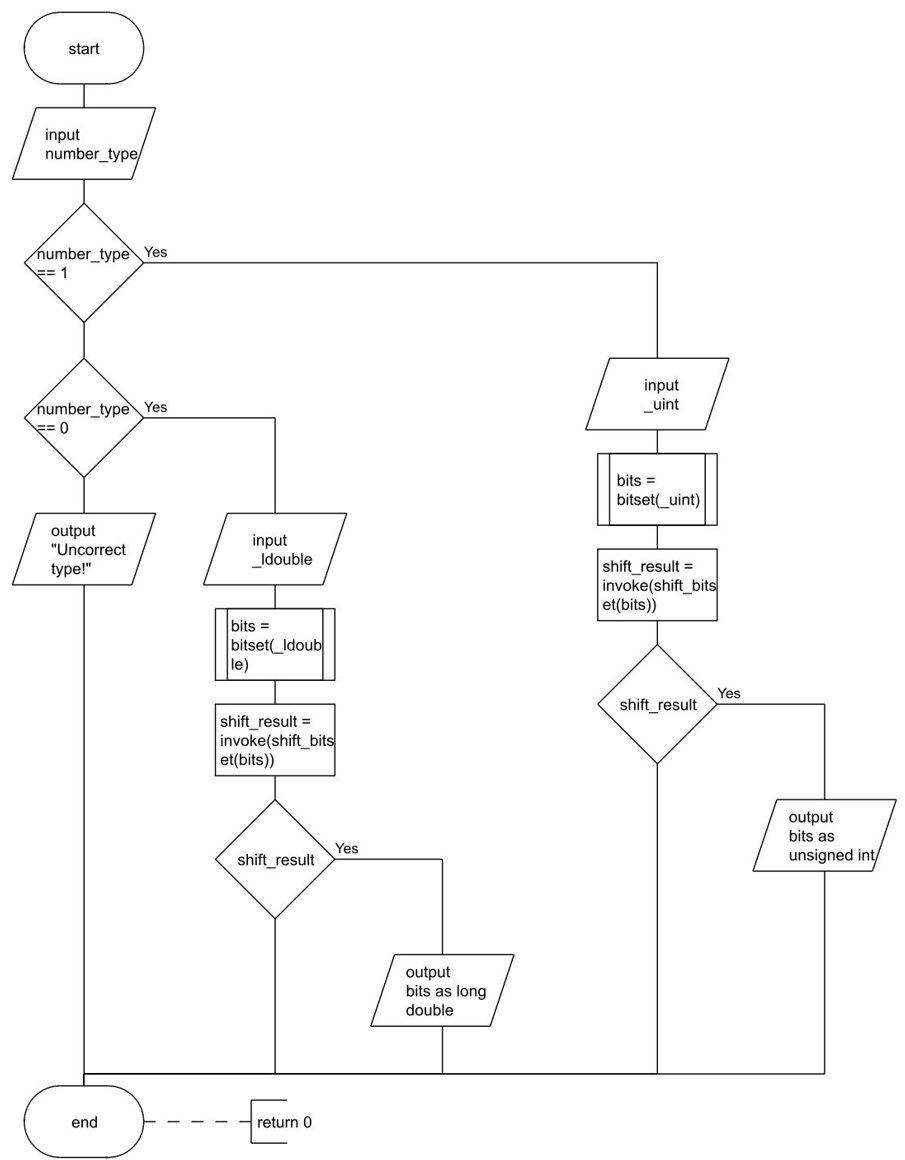
cout << (\*source\_bitset) << endl;

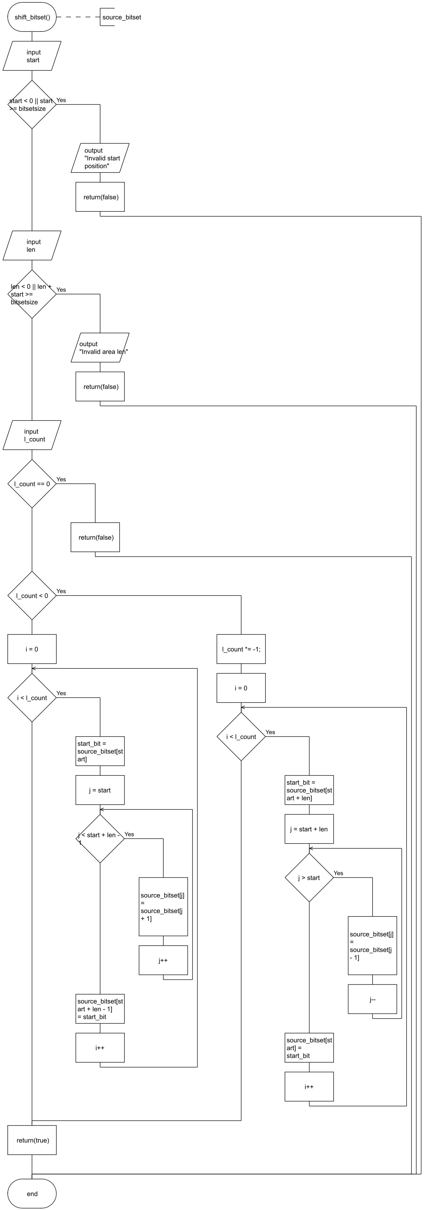
return true;

}

**Блок-схема алгоритма**

**main**

**Shift\_bitset**

****

**Примеры выполнения программы**

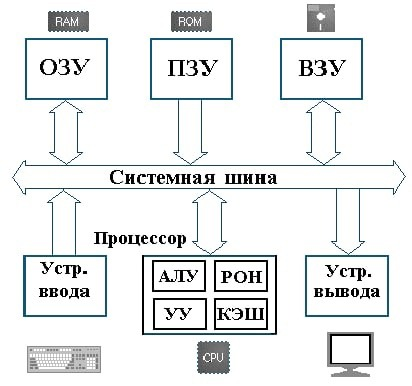
**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Cтруктурная схема аппаратных средств**



**Выводы.**

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