Министерство образования и науки Российской Федерации   
Федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра «Информационные технологии и автоматизированные системы

**Отчёт   
по лабораторной работе №11.2**

Дисциплина: основы алгоритмизации и программирования

Тема: «Очереди»

Вариант №14

Выполнил работу:  
студент группы РИС-20-2б  
Вичугов Алексей Дмитриевич

Проверила:  
доцент кафедры ИТАС  
Полякова Ольга Андреевна

Пермь 2021

**Цели работы**

Получить практические навыки работы с очередями;

**Постановка задачи**

1. Сформировать очередь. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

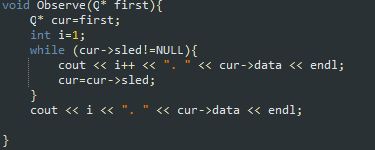
5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

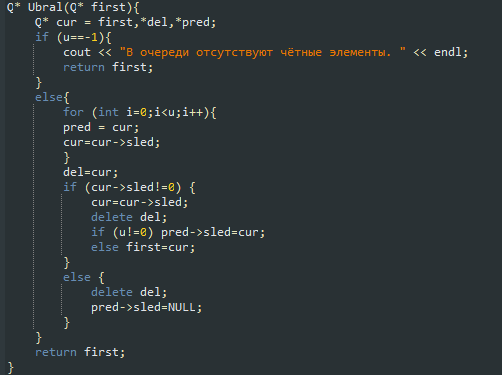
1. Для решения задачи необходимо:
   1. Сформировать очередь:



* 1. Создать функцию, распечатывающую созданную очередь:



* 1. Удалить последний чётный элемент:



1. В ходе работы были использованы типы данных:
   1. int:  
       int n,u,i;
   2. Указатели на тип данных Q:

Q \*cur,\*first,\*del,\*pred,\*o;

1. Данные для решения задачи представлены в виде структуры:  
   struct Q{

int data;

Q\* sled;

};

1. В ходе работы программы использовалась структура данных Q:

struct Q{

int data;

Q\* sled;

};

1. Ввод данных в переменны производился с помощью оператора cin:  
   cin >> n;  
   cin >> cur->data;
2. При решении поставленных задач использовались циклы:
   1. Для ввода данных и проверки правильности ввода цикл while:

while (cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Неверный ввод! Элемент должен являться целым числом! Повторите ввод: " << endl;}

**Полный код программы**

#include <iostream>

using namespace std;

struct Q{

int data;

Q\* sled;

};

int u=-1;

Q\* Create(){

int n;

Q\* cur, \*first,\*nov;

first = new Q;

first->sled=NULL;

cur=first;

cout << "Количество элементов очереди: " << endl;

cin >> n;

while (n<2||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Неверный ввод! Количество элементов должно являться числом и иметь значение не менее 2! Повторите ввод: " << endl;

cin >> n;

}

cout << "Значение 1-го элемента: " << endl;

cin >> first->data;

while (cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Неверный ввод! Элемент должен являться целым числом! Повторите ввод: " << endl;

cin >> first->data;;

}

if (first->data%2==0) u=0;

for (int i=1;i<n;i++){

nov= new Q;

cur->sled=nov;

cur=nov;

cur->sled=NULL;

cout << "Значение " << i+1 << "-го элемента: " << endl;

cin >> cur->data;

while (cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Неверный ввод! Элемент должен являться целым числом! Повторите ввод: " << endl;

cin >> cur->data;

}

if (cur->data%2==0) u=i;

}

return first;

}

void Observe(Q\* first){

Q\* cur=first;

int i=1;

while (cur->sled!=NULL){

cout << i++ << ". " << cur->data << endl;

cur=cur->sled;

}

cout << i << ". " << cur->data << endl;

}

Q\* Ubral(Q\* first){

Q\* cur = first,\*del,\*pred;

if (u==-1){

cout << "В очереди отсутствуют чётные элементы. " << endl;

return first;

}

else{

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

pred = cur;

cur=cur->sled;

}

del=cur;

if (cur->sled!=0) {

cur=cur->sled;

delete del;

if (u!=0) pred->sled=cur;

else first=cur;

}

else {

delete del;

pred->sled=NULL;

}

}

return first;

}

void Set(){

const int fon = system("Color F0");

system ("chcp 1251>nul");

}

int main(){

Set();

Q\* o = Create();

Observe(o);

o=Ubral(o);

cout << "Обновлённая очередь: " << endl;

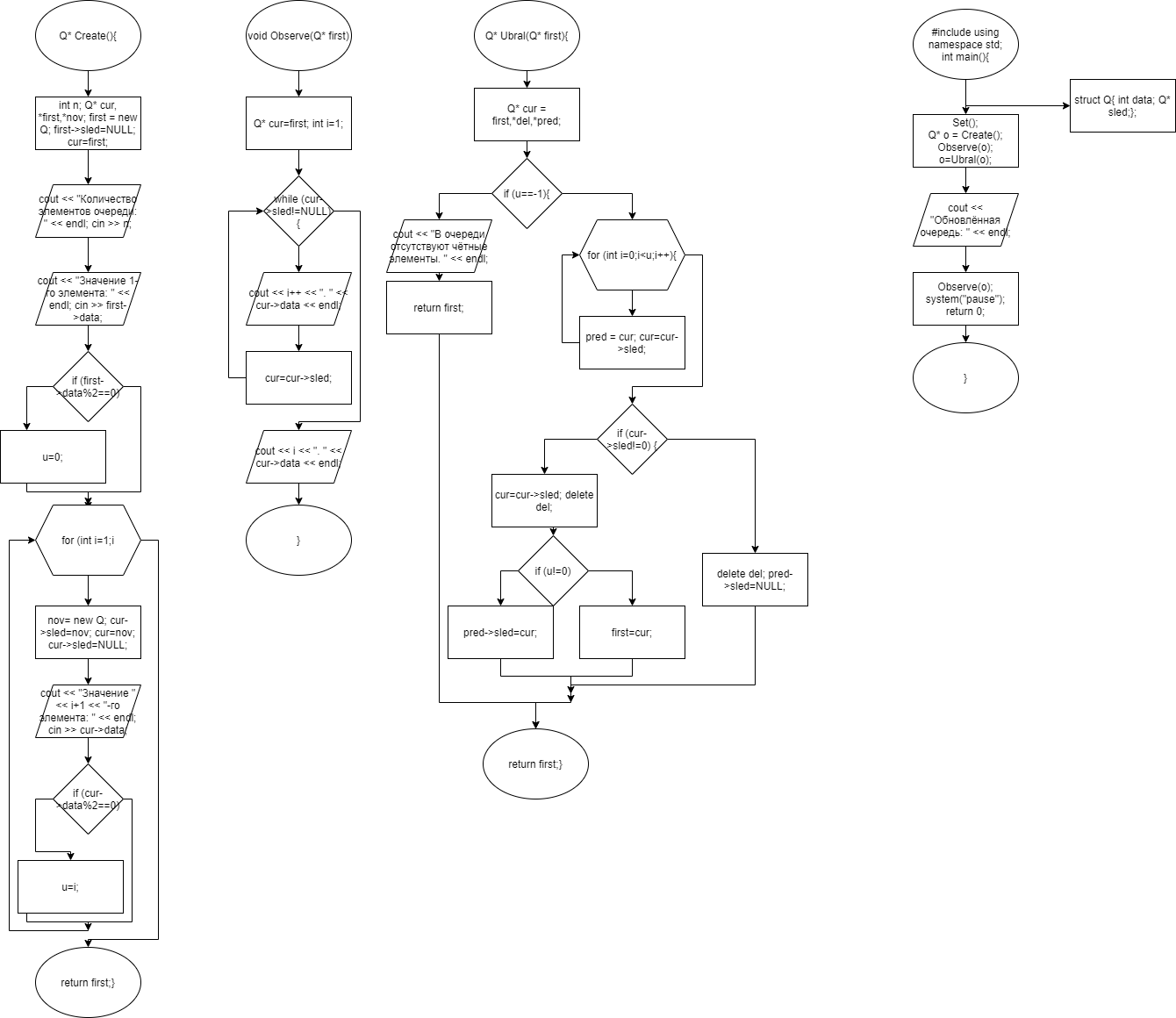
Observe(o);

system("pause");

return 0;

}

**Блок-схема**



**Результаты тестов**

