МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 3 по курсу «Алгоритмы и структуры данных»**

«Реализация линейной структуры данных «Очередь» и основные

алгоритмы обработки.»

Вариант №13

ВЫПОЛНИЛ студент группы 21-ИТ-1

Макеёнок Д.И.

ПРОВЕРИЛ преподаватель

Виноградова А.Д.

Полоцк, 2022 г.

**Цель работы:** ознакомиться с основами линейной структуры данных

«Очередь», изучить основные алгоритмы обработки ЛСД «Очередь», научиться применять полученные знания на практике.

**Теоретические сведения:**

1. Определение понятия очередь.

Очередью называется упорядоченный набор элементов, которые

могут удаляться с её начала и помещаться в её конец.

1. Опишите принцип работы очереди.

Очередь организована согласно дисциплине обслуживания FIFO:

• FIRST — первый

• INPUT — вошел

• FIRST — первый

• OUTPUT — вышел

1. Операции для работы с очередью.

• init() инициализация очереди.

• insert (q, x) — помещение элемента x в конец очереди q (q —

указатель на очередь);

• x=remove (q) — удаление элемента x из очереди q;

• isempty(q) — возвращает 1, если очередь пуста и 0 в противном

случае;

• print(q) – вывод элементов очереди q.

1. Перечислите способы реализации очереди.

• с помощью одномерного массива;

• с помощью связанного списка;

• с помощью класса объектно-ориентированного программирования.

**Код программы:**

**1.**

#include <iostream>

#include <vector>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus"); //русский язык

vector<int> queue; //создаётся массив

int mem = 0, d = 0, a, n;

cout << "Введите количество элементов: " << endl;

cin >> n;

cout << "Введите элементы: " << endl;

for (int i = 0; n != i; i++) {

cin >> a;

queue.push\_back(a); //заполнение очереди символами

if (a > mem) { //сравнение элементов массива

mem = a; //запоминается самый большое число

}

}

bool del = false;

for (int i = 0; n != i; i++) {

if (del) {

queue[i] = ' ';

}

else { //выводятся все числа до тех пор пока не доходит до самого большого

if (queue[i] == mem) { //все остальные символы удаляются

del = true;

cout << queue[i] << " ";

}

else cout << queue[i] << " ";

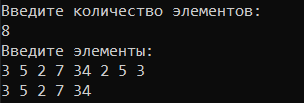
}

}

return 0;

}

**Результат работы программы:**



**2.**

#include<iostream>

using namespace std;

// структура

typedef struct node

{

int data;

// Меньшее числовое значение означает большее значение

int priority;

struct node\* next;

} Node;

// Функция создания нового узла

Node\* newNode(int d, int p)

{

Node\* temp = (Node\*)malloc(sizeof(Node));

temp->data = d;

temp->priority = p;

temp->next = NULL;

return temp;

}

// Функция возврата значения корня

int peek(Node\*\* head)

{

return (\*head)->data;

}

// Функция удаление узла с наибольшим приоритетомRemoves

void pop(Node\*\* head)

{

Node\* temp = \*head;

(\*head) = (\*head)->next;

free(temp);

}

// Функция добавления узла в соответствии с приоритетом

void push(Node\*\* head, int d, int p)

{

Node\* start = (\*head);

// Создание нового узла

Node\* temp = newNode(d, p);

// замена head node.

if ((\*head)->priority > p)

{

// Добавление нового узла перед корнем листа

temp->next = \*head;

(\*head) = temp;

}

else

{

while (start->next != NULL &&

start->next->priority < p)

{

start = start->next;

}

temp->next = start->next;

start->next = temp;

}

}

// Функция проверки, пуст ли список

int isEmpty(Node\*\* head)

{

return (\*head) == NULL;

}

int main()

{

setlocale(LC\_ALL, "Russian");

int num, pre, n;

cout << "Введите количество элементов: ";

cin >> n;

cout << "Введите элементы: ";

cin >> num >> pre;

// Создание очереди с приоритетом

Node\* pq = newNode(num, pre);

for (int i = 0; i < n - 1; i++) {

cin >> num >> pre;

push(&pq, num, pre);

}

while (!isEmpty(&pq))

{

cout << " " << peek(&pq);

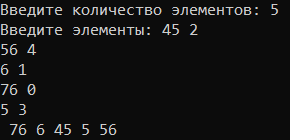
pop(&pq);

}

return 0;

}

**Результат работы программы:**

****

**Вывод:** я ознакомился с основами линейной структуры данных

«Очередь», изучил основные алгоритмы обработки ЛСД «Очередь», научился применять полученные знания на практике.