Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Очереди.

Выполнил работу

Студент группы РИС-22-1Б

Лихачев Д.А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь-2023

**Постановка задачи**

Реализовать метод очередей. Требуется написать функции добавления и удаления элементов, а также дополнительные функции и методы очередей.

**Код программы**

#include<iostream>

#define\_SIZE 1000

usingnamespace std;

classQueue

{

int\* arr;

int capacity;

int front;

int rear;

int count;

public:

Queue(intsize = \_SIZE);

~Queue();

int dequeue();

void enqueue(intx);

int peek();

int size();

bool isEmpty();

bool isFull();

};

Queue::Queue(intsize)

{

arr = newint[size];

capacity = size;

front = 0;

rear = -1;

count = 0;

}

Queue::~Queue() {

delete[] arr;

}

intQueue::dequeue()

{

if (isEmpty())

{

cout <<"Underflow\nProgram Terminated\n";

exit(EXIT\_FAILURE);

}

int x = arr[front];

cout <<"Removing "<< x << endl;

front = (front + 1) % capacity;

count--;

return x;

}

voidQueue::enqueue(intitem)

{

if (isFull())

{

cout <<"Overflow\nProgram Terminated\n";

exit(EXIT\_FAILURE);

}

cout <<"Inserting "<<item<< endl;

rear = (rear + 1) % capacity;

arr[rear] = item;

count++;

}

intQueue::peek()

{

if (isEmpty())

{

cout <<"Underflow\nProgram Terminated\n";

exit(EXIT\_FAILURE);

}

return arr[front];

}

intQueue::size()

{

return count;

}

boolQueue::isEmpty()

{

return (size() == 0);

}

boolQueue::isFull()

{

return (size() == capacity);

}

int main()

{

Queue q(5);

q.enqueue(1);

q.enqueue(2);

q.enqueue(3);

cout << "The front element is " << q.peek() << endl;

q.dequeue();

q.enqueue(4);

cout << "The queue size is " << q.size() << endl;

q.dequeue();

q.dequeue();

q.dequeue();

if (q.isEmpty())

cout << "The queue is empty\n";

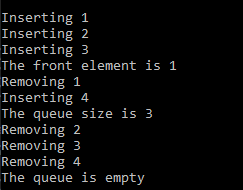
else

cout << "The queue is not empty\n";

return 0;

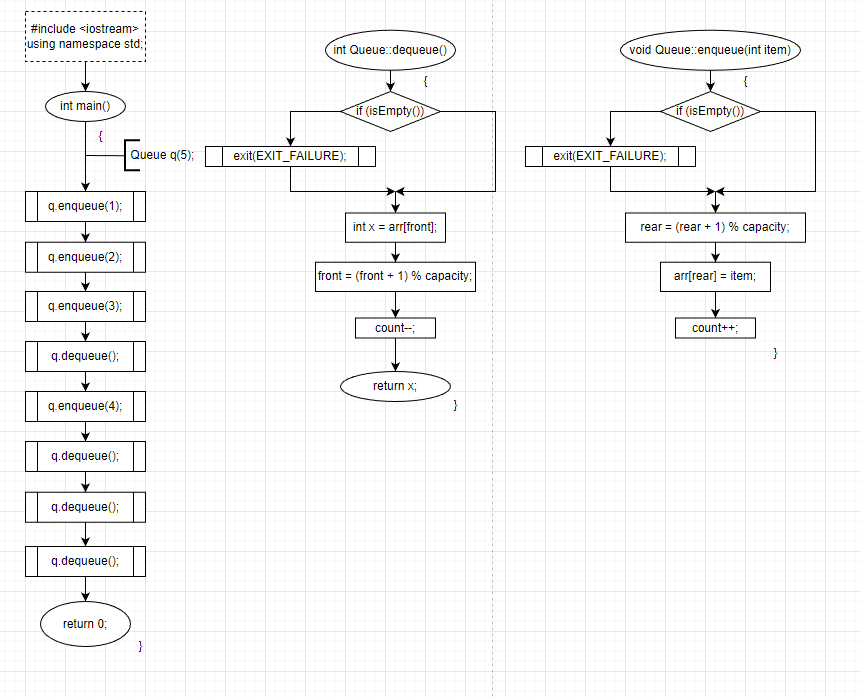
}

**Вывод программы**

****

***Рисунок 1 – вывод программы***

**Блок-схема**

****

***Рисунок 3 – Схема алгоритма программы***

**Анализ результатов**

Программа выполняет свою задачу.