

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Лабораторна робота № 4
з дисципліни «Алгоритми та структури даних»
На тему: «Структури даних: стеки, черги»

Виконав:
студент групи ФІ-12
Завалій Олександр

Реалізація завдання

Варіант №5

Реалізувати чергу двома способами:

1. Масивом, використавши кільцеву чергу.
2. Зв'язним списком.

Task I

Масив, кільцева черга.

```
#include <iostream>
using namespace std;
#define SIZE 6

template <typename T>
class QueueRing {
private:
    T arr[SIZE];
    int front, rear;
public:
    QueueRing() {
        front = rear = -1;
    }

    bool isEmpty();
    bool isFull();
    void enqueue(int num);
    void dequeue();
    void print();
};

template <typename T>
bool QueueRing<T>::isEmpty() {
    return front == -1 and rear == -1;
}

template <typename T>
bool QueueRing<T>::isFull() {
    return (rear + 1) % SIZE == front;
}

template <typename T>
void QueueRing<T>::enqueue(int value) {
    if (isFull()) {
        cout << "Queue is full\n";
        return;
    } else if (isEmpty()) {
        front = rear = 0;
    } else {
        rear = (rear + 1) % SIZE;
    }
    arr[rear] = value;
}
```

```

template <typename T>
void QueueRing<T>::dequeue() {
    if (isEmpty()) {
        cout << "Queue is empty\n";
        return;
    } else if (front == rear) {
        front = rear = -1;
    } else {
        front = (front + 1) % SIZE;
    }
}

template <typename T>
void QueueRing<T>::print() {
    if (isEmpty()) {
        cout << "Queue is empty\n";
        return;
    }
    cout << "Queue: ";
    for (int i = front; i != rear; i = (i + 1) % SIZE) {
        cout << arr[i] << ' ';
    }
    cout << arr[rear] << endl;
}

int main() {
    QueueRing<char> q;

    for (int i = 97; i < SIZE+97; i++) {
        q.enqueue(i);
    }
    q.print();

    q.enqueue(102);

    for (int i = 0; i < 4; i++) {
        q.dequeue();
    }
    q.print();

    q.enqueue(103);
    q.enqueue(104);
    q.enqueue(105);
    q.print();

    return 0;
}

```

Task II

Зв'язний список.

```
#include <iostream>
#include <string>
using namespace std;

template <typename T>
class Queue {
private:
    struct Node {
        T data;
        Node* next;
    };
    Node* front;
    Node* rear;
public:
    Queue() {
        front = rear = NULL;
    }
    ~Queue() {
        Node* current = front;
        while (current) {
            Node* temp = current->next;
            delete current;
            current = temp;
        }
    }

    void enqueue(T value);
    void dequeue();
    void dequeueA();
    void print();
};

template <typename T>
void Queue<T>::enqueue(T value) {
    Node* tempnode = new Node();
    tempnode->data = value;
    tempnode->next = NULL;
    if (front == NULL and rear == NULL){
        front = rear = tempnode;
    }
    rear->next = tempnode;
    rear = tempnode;
}

template <typename T>
void Queue<T>::dequeue(){
    Node* tempnode = front;
    if (!tempnode) {
        cout << "Queue is empty!" << endl;
        return;
    }
    if (tempnode == rear) {
        front = rear = NULL;
    } else {
        front = tempnode->next;
    }
    delete tempnode;
}
```

```

template <typename T>
void Queue<T>::dequeueA(){
    Node* tempnode = front;
    while (tempnode) {
        Node* temp = tempnode->next;
        delete tempnode;
        tempnode = temp;
    }
    front = rear = NULL;
}

template <typename T>
void Queue<T>::print(){
    Node* tempnode = front;
    if (!tempnode) {
        cout << "Queue is empty!" << endl;
        return;
    }
    while (tempnode) {
        cout << tempnode->data << ' ';
        tempnode = tempnode->next;
    }
    cout << endl;
}

int main() {
    Queue<string> qlist;
    string temp;

    do {
        cin >> temp;
        if (temp != "q"){
            qlist.enqueue(temp);
        } else {
            break;
        }
    } while(true);

    qlist.print();
    qlist.dequeue();
    qlist.print();

    qlist.enqueue("END");
    qlist.print();

    qlist.dequeueA();
    qlist.print();

    return 0;
}

```