Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Реализация дека на основе массива»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Проверил:

Белодед Николай Иванович

2024, Минск

#include <iostream>

#include <Windows.h>

#include <string>

#include <regex>

using namespace std;

regex checknum("[\\-]?[0-9]+"); // регулярное выражение для проверки корректности ввода

const int MAX\_SIZE = 1000; // максимальный размер дека

class Deque { // класс дека

private: // приватная область

int arr[MAX\_SIZE]; // массив для хранения элементов дека

int size; // текущий размер дека

public: // публичная область

Deque(){ // конструктор

size = 0; // размер равен 0

}

bool isEmpty() { // функция проверки на пустоту

return (size == 0); // проверка равенства текущего размера 0

}

bool isFull() { // функция проверки на переполненность

return (size == MAX\_SIZE - 1); // проверка равенства текущего размера максимальному

}

void pushfront(int val) { // функция добавления элемента слева

if (isFull()) { // проверяем стек на заполненность

cout << "Дек переполнен" << '\n';

return;

}

for (int i = size - 1; i >= 0; i--) { // в цикле

arr[i + 1] = arr[i]; // сдвигаем все элементы на 1 вправо

}

size++; // увеличиваем текущий размер

arr[0] = val; // присваиваем первому элементу новое значение

}

void pushback(int val) {// функция добавления элемента справа

if (isFull()) {// проверяем стек на заполненность

cout << "Дек переполнен" << '\n';

return;

}

size++;// увеличиваем текущий размер

arr[size - 1] = val;// присваиваем последнему элементу новое значение

}

void popfront() { // функция удаления элемента слева

if (isEmpty()) { // проверка дека на пустоту

cout << "Дек пуст" << '\n';

return;

}

cout << "Удалено значение " << arr[0] << " из начала дека." << '\n';

for (int i = 1; i < size; i++) { // в цикле

arr[i - 1] = arr[i]; // сдвигаем элементы влево

}

size--; // уменьшаем текущий размер

}

void popback() {

if (isEmpty()) {// проверка дека на пустоту

cout << "Дек пуст" << '\n';

return;

}

cout << "Удалено значение " << arr[size - 1] << " из конца дека." << '\n';

size--;// уменьшаем текущий размер

}

int getfront() { // функция получения значения первого слева элемента

if (isEmpty()) {// проверка дека на пустоту

cout << "Дек пуст" << '\n';

return -1;

}

// иначе

return arr[0]; // возвращаем нужное значение

}

int getback() {

if (isEmpty()) {// проверка дека на пустоту

cout << "Дек пуст" << '\n';

return -1;

}

// иначе

return arr[size - 1];// возвращаем нужное значение

}

void output() { // функция вывода всех элементов дека

if (isEmpty()) {// проверка дека на пустоту

cout << "Дек пуст" << '\n';

return;

}

for (int i = 0; i < size; i++) { // в цикле

cout << arr[i] << ' '; // выводим все элементы

}

cout << '\n';

}

};

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

Deque deq; // создаем объект класса Deque

char choice; // переменная для выбора опции

string c; // буферная строка

do {

cout << "Введите:\n";

cout << "1 - для добавления элемента в дек слева\n";

cout << "2 - для добавления элемента в дек справа\n";

cout << "3 - для удаления элемента из дека слева\n";

cout << "4 - для удаления элемента из дека справа\n";

cout << "5 - для проверки на пустоту\n";

cout << "6 - для вывода первого элемента\n";

cout << "7 - для вывода последнего элемента\n";

cout << "8 - для вывода всех элементов дека\n";

cout << "0 - для выхода\n";

getline(cin, c);

while (c == "") {

getline(cin, c);

}

while (c.length() != 1) {

cout << "Некорректный ввод\n";

getline(cin, c);

}

choice = c[0];

switch (choice) {

case '1': {

int n;

cout << "Введите число, которое хотите добавить в дек\n";

string b;

getline(cin, b);

while (!regex\_match(b, checknum)) {

cout << "Некорректный ввод\n";

getline(cin, b);

}

n = stoi(b);

deq.pushfront(n);

break;

}

case '2': {

int n;

cout << "Введите число, которое хотите добавить в дек\n";

string b;

getline(cin, b);

while (!regex\_match(b, checknum)) {

cout << "Некорректный ввод\n";

getline(cin, b);

}

n = stoi(b);

deq.pushback(n);

break;

}

case '3': {

deq.popfront();

break;

}

case '4': {

deq.popback();

break;

}

case '5': {

if (!deq.isEmpty()) {

cout << "Дек не пуст\n";

}

else {

cout << "Дек пуст\n";

}

break;

}

case '6': {

int k = deq.getfront();

if (k != -1) {

cout << k << '\n';

}

break;

}

case'7': {

int k = deq.getback();

if (k != -1) {

cout << k << '\n';

}

break;

}

case'8': {

deq.output();

}

case'0': {

break;

}

default: {

cout << "Некорректный ввод\n";

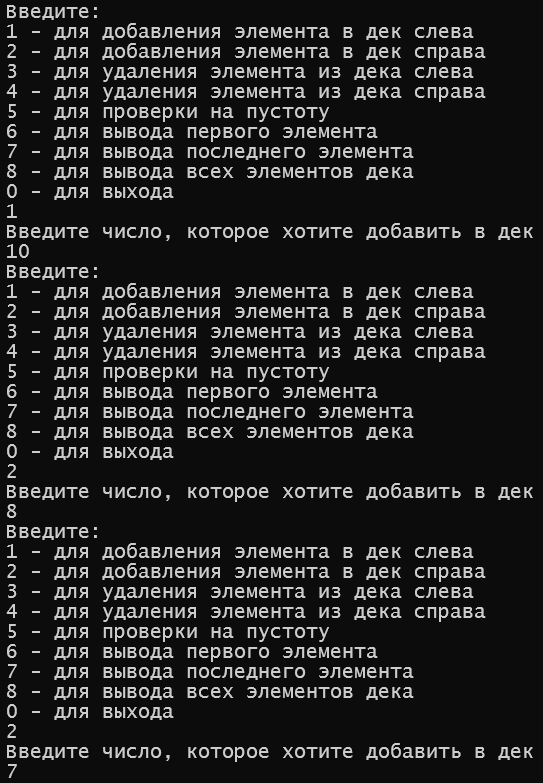
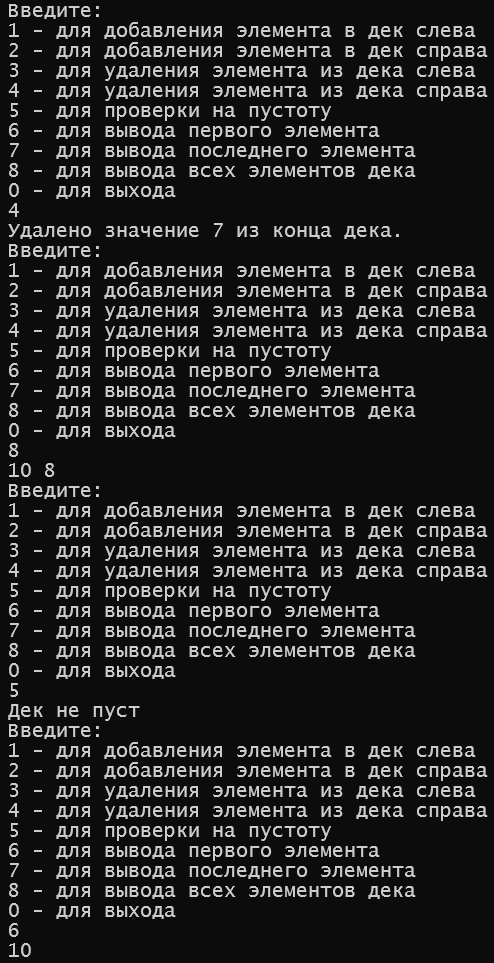
break;

}

}

} while (choice != '0');

}

Результат выполнения:

