Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Динамические списки»

Выполнили:

студенты группы 21ВВ1.3

Симонов Михаил

Шуравин Александр

Приняли:

д.т.н., профессор Митрохин М.А.

К.т.н., доцент Юрова О. В.

Пенза 2022

**Название**

Динамические списки.

**Лабораторное задание.**

Задание 1. Реализовать приоритетную очередь, путём добавления элемента в список в соответствии с приоритетом объекта (т.е. объект с большим приоритетом становится перед объектом с меньшим приоритетом).

Задание 2. На основе приведенного кода реализуйте структуру данных *Очередь.*

Задание 3. На основе приведенного кода реализуйте структуру данных *Стек.*

**Листинг**

laba3.1.cpp

#include "stdafx.h"

#include <iostream>

struct node

{

char inf[256];

int P;

struct node\* prev;

};

struct node\* head = NULL, \* last = NULL;

void push(const char\* str, int P);

void pop();

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

char com[10], data[256];

int pr;

while (1) {

if (head != NULL) {

node\* tmp = head;

cout << "Очередь: ";

while (1) {

cout << tmp->inf << " ";

if (tmp == last) {

cout << endl << endl;

break;

}

tmp = tmp->prev;

}

}

cout << "Введите команду push или pop для работы с очередью. Чтобы выйти введите exit: ";

cin >> com;

cout << endl;

if (!strcmp(com, "push")) {

cout << "Введите данные для помещения в очередь: ";

cin >> data;

cout << "Введите номер для приоритета в очереди: ";

cin >> pr;

cout << endl;

push(data, pr);

}

else if (!strcmp(com, "pop")) {

pop();

}

else if (!strcmp(com, "exit")) {

exit(0);

}

else {

cout << "Вы ввели команду неверно" << endl << endl;

}

}

}

void push(const char\* str, int P) {

node\* tmp = (node\*)malloc(sizeof(node));

if (head == NULL) {

head = tmp;

last = tmp;

tmp->P = P;

strcpy\_s(tmp->inf, str);

}

else {

char flag = 0;

tmp->P = P;

strcpy\_s(tmp->inf, str);

node\* sravn = head, \* sravnprev = head;

while (flag == 0) {

while (1) {

if (sravn->P == tmp->P) {

if (sravn == last) {

last = tmp;

sravn->prev = tmp;

flag = 1;

break;

}

else {

tmp->prev = sravn->prev;

sravn->prev = tmp;

flag = 1;

break;

}

}

else if (sravn->P < tmp->P) {

if (sravn == last) {

last = tmp;

sravn->prev = tmp;

flag = 1;

break;

}

else {

sravnprev = sravn;

sravn = sravn->prev;

break;

}

}

else if (sravn->P > tmp->P) {

if (sravn == head) {

tmp->prev = head;

head = tmp;

flag = 1;

break;

}

else {

tmp->prev = sravn;

sravnprev->prev = tmp;

flag = 1;

break;

}

}

}

}

}

}

void pop() {

if (head == NULL) {

cout << "Очередь пустая" << endl << endl;

return;

}

else if (head == last) {

cout << "Извлечённый элемент: " << head->inf << endl << endl;

free(head);

head = NULL;

last = NULL;

}

else {

cout << "Извлечённый элемент: " << head->inf << endl;

node\* tmp = head;

head = head->prev;

free(tmp);

}

}

laba3.2.cpp

#include "stdafx.h"

#include <iostream>

struct node

{

char inf[256];

struct node\* prev;

};

struct node\* head = NULL, \* last = NULL;

void push(const char\* str);

void pop();

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

char com[10], data[256];

while (1) {

if (head != NULL) {

node\* tmp = head;

cout << "Очередь: ";

while (1) {

cout << tmp->inf << " ";

if (tmp == last) {

cout << endl << endl;

break;

}

tmp = tmp->prev;

}

}

cout << "Введите команду push или pop для работы с очередью. Чтобы выйти введите exit: ";

cin >> com;

cout << endl;

if (!strcmp(com, "push")) {

cout << "Введите данные для помещения в очередь: ";

cin >> data;

cout << endl;

push(data);

}

else if (!strcmp(com, "pop")) {

pop();

}

else if (!strcmp(com, "exit")) {

exit(0);

}

else {

cout << "Вы ввели команду неверно" << endl << endl;

}

}

}

void push(const char\* str) {

node\* tmp = (node\*)malloc(sizeof(node));

if (head == NULL) {

head = tmp;

last = tmp;

strcpy\_s(last->inf, str);

}

else {

last->prev = tmp;

strcpy\_s(tmp->inf, str);

last = tmp;

}

}

void pop() {

if (head == NULL) {

cout << "Очередь пустая" << endl << endl;

return;

}

else if (head == last) {

cout << "Извлечённый элемент: " << head->inf << endl << endl;

free(head);

head = NULL;

last = NULL;

}

else {

cout << "Извлечённый элемент: " << head->inf << endl;

node\* tmp = head;

head = head->prev;

free(tmp);

}

}

laba3.3.cpp

#include "stdafx.h"

#include <iostream>

struct node

{

char inf[256];

struct node\* next;

};

struct node\* last = NULL;

void push(const char\* str);

void pop();

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

char com[10], data[256];

while (1) {

if (last != NULL) {

node\* tmp = last;

cout << "Стэк: ";

while (1) {

cout << tmp->inf << " ";

if (tmp->next == NULL) {

cout << endl << endl;

break;

}

tmp = tmp->next;

}

}

cout << "Введите команду push или pop для работы со стеком. Чтобы выйти введите exit: ";

cin >> com;

cout << endl;

if (!strcmp(com, "push")) {

cout << "Введите данные для помещения в стэк: ";

cin >> data;

cout << endl;

push(data);

}

else if (!strcmp(com, "pop")) {

pop();

}

else if (!strcmp(com, "exit")) {

exit(0);

}

else {

cout << "Вы ввели команду неверно" << endl;

}

}

}

void push(const char\* str) {

node\* tmp = (node\*)malloc(sizeof(node));

if (last == NULL) {

last = tmp;

strcpy\_s(last->inf, str);

last->next = NULL;

}

else {

tmp->next = last;

strcpy\_s(tmp->inf, str);

last = tmp;

}

}

void pop() {

if (last == NULL) {

cout << "Стэк пустой" << endl << endl;

return;

}

else if (last->next == NULL) {

cout << "Извлечённый элемент: " << last->inf << endl << endl;

free(last);

last = NULL;

}

else {

cout << "Извлечённый элемент: " << last->inf << endl;

node\* tmp = last;

last = last->next;

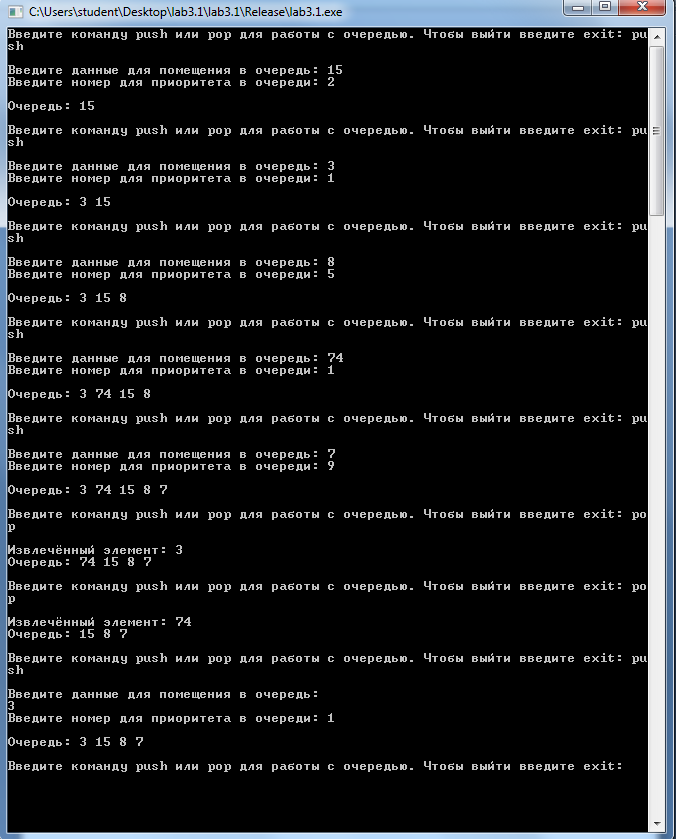
free(tmp);

}

}

**Результат работы программы**

Результаты работы программы (задание 1) показаны на рисунке 1.



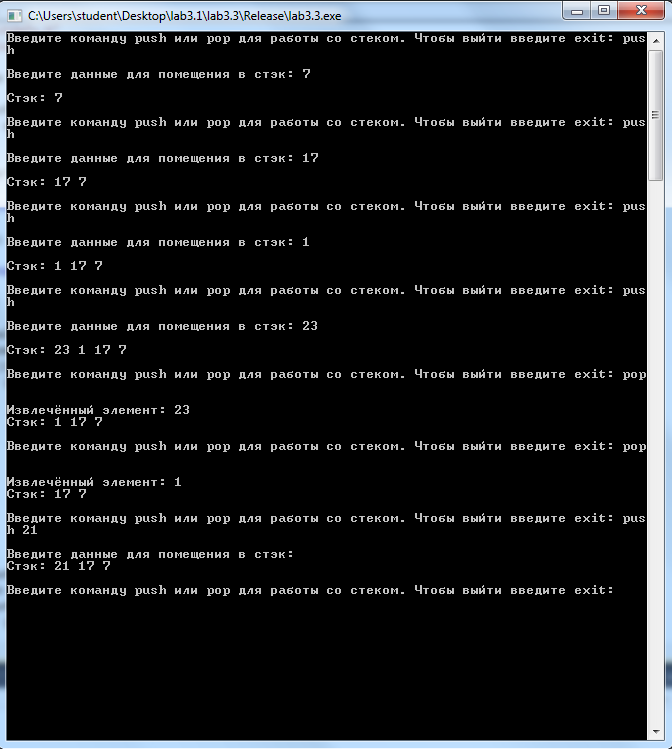
**Рисунок 1 — Результаты работы программы**

Результаты работы программы (задание 2) показаны на рисунке 2.



**Рисунок 2 — Результаты работы программы**

Результаты работы программы (задание 3) показаны на рисунке 3.



**Рисунок 3 — Результаты работы программы**

### Вывод

Список – это абстрактный тип данных. Реализовать его можно с помощью массивов: динамического и статического. Каждый элемент (иногда кроме первого и последнего) динамического списка имеет указатель, который связывает его с другим элементом. Так же для работы с динамическим списком необходимо хранить “заглавный” элемент списка, чтобы отталкиваясь от него работать со списком.