Министерство 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 21ВВ4:

Шутихин А.Э.

Ошкин И.В

Приняли:

Юрова О. В.

Акифьев И.В.

Пенза 2022

**Лабораторное задание:**

1) Реализовать приоритетную очередь, путём добавления элемента в список в соответствии с приоритетом объекта

2\*) На основе приведенного кода реализуйте структуру данных Очередь.

3\*) На основе приведенного кода реализуйте структуру данных Стек.

**Задание 1**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale>

struct node

{

char inf[256];

int val;

struct node\* prev; // ссылка на предыдущий элемент

};

struct node\* head = NULL, \* last = NULL; // указатели на первый и последний элементы списка

void in(const char\* str, int val) {

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

if (head == NULL) {

head = list;

last = list;

list->val = val;

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

}

else {

char flag = 0;

list->val = val;

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

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

while (flag == 0) {

while (1) {

if (sravn->val == list->val) { //если номера равны

if (sravn == last) { //если последний элемент равен вводимому

last = list;

sravn->prev = list;

flag = 1;

break;

}

else {

list->prev = sravn->prev;

sravn->prev = list;

flag = 1;

break;

}

}

else if (sravn->val < list->val) { //если номер с меньшим приоритетом

if (sravn == last) {

last = list;

sravn->prev = list;

flag = 1;

break;

}

else {

sravnprev = sravn;

sravn = sravn->prev;

break;

}

}

else if (sravn->val > list->val) { //если номер с большим приоритетом

if (sravn == head) {

list->prev = head;

head = list;

flag = 1;

break;

}

else {

list->prev = sravn;

sravnprev->prev = list;

flag = 1;

break;

}

}

}

}

}

}

void out() {

if (head == NULL) {

("Очередь пустая\n");

return;

}

else if (head == last) {

printf("Извлечённый элемент: %s\n",head->inf);

free(head);

head = NULL;

last = NULL;

}

else {

printf("Извлечённый элемент: %s\n", head->inf);

node\* tmp = head;

head = head->prev;

free(tmp);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

char data[256];

int pr;

int buf;

while (1) {

if (head != NULL) {

node\* list = head;

printf("Очередь: ");

while (1) {

printf("%s ", list->inf);

if (list == last) {

printf("\n");

break;

}

list = list->prev;

}

}

printf("Введите: 1 для помещения в очередь, 2 для извлечения из очереди, 0 для выхода ");

scanf("%d", &buf);

printf("\n");

if (buf == 1) {

printf("Введите данные для помещения в очередь: ");

scanf("%s", data);

printf("Введите номер для приоритета в очереди: ");

scanf\_s("%d", &pr);

printf("\n");

in(data, pr);

}

else if (buf == 2) {

out();

}

else if (buf == 0) {

exit(0);

}

else {

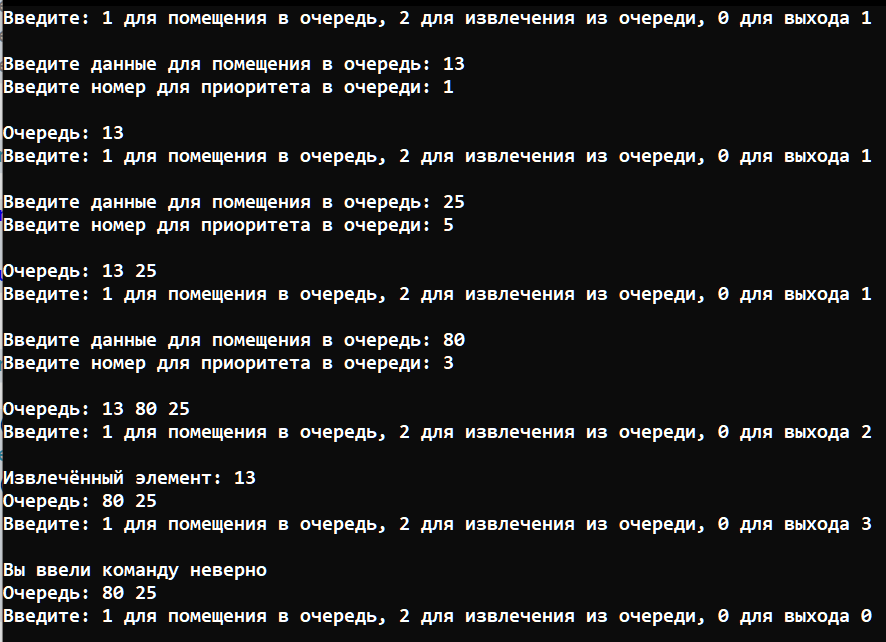
printf("Вы ввели команду неверно\n");

}

}

}

**Результат:**

****

**Задание 2**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale>

struct node

{

char inf[256];

struct node\* prev; // ссылка на предыдущий элемент

};

struct node\* head = NULL, \* last = NULL; // указатели на первый и последний элементы списка

void in(const char\* str) {

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

if (head == NULL) {

head = list;

last = list;

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

}

else {

last->prev = list;

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

last = list;

}

}

void out() {

if (head == NULL) {

("Очередь пустая\n");

return;

}

else if (head == last) {

printf("Извлечённый элемент: %s\n",head->inf);

free(head);

head = NULL;

last = NULL;

}

else {

printf("Извлечённый элемент: %s\n", head->inf);

node\* tmp = head;

head = head->prev;

free(tmp);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

char data[256];

int buf;

while (1) {

if (head != NULL) {

node\* list = head;

printf("Очередь: ");

while (1) {

printf("%s ", list->inf);

if (list == last) {

printf("\n");

break;

}

list = list->prev;

}

}

printf("Введите: 1 для помещения в очередь, 2 для извлечения из очереди, 0 для выхода ");

scanf("%d", &buf);

printf("\n");

if (buf == 1) {

printf("Введите данные для помещения в очередь: ");

scanf("%s", data);

printf("\n");

in(data);

}

else if (buf == 2) {

out();

}

else if (buf == 0) {

exit(0);

}

else {

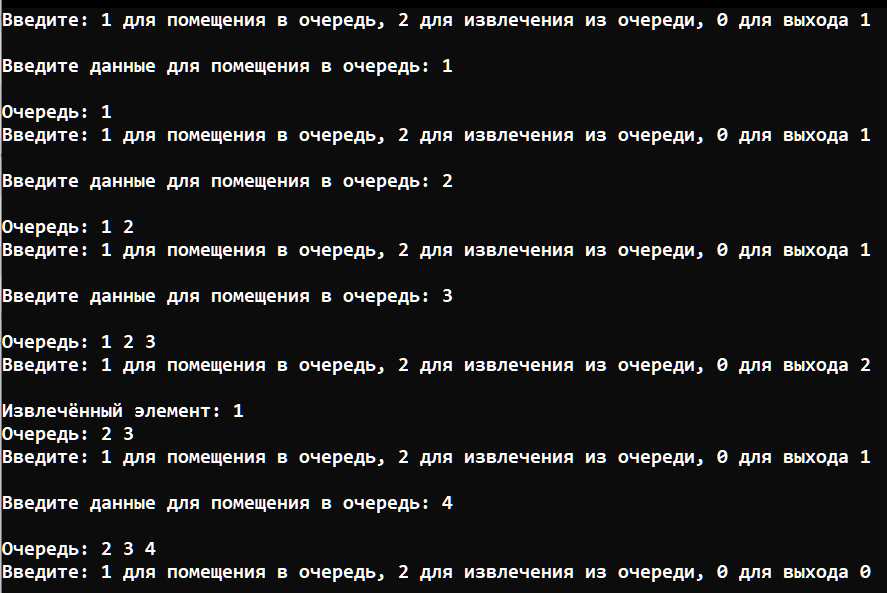
printf("Вы ввели команду неверно\n");

}

}

}

**Результат:**

****

**Задание 3**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale>

struct node

{

char inf[256];

struct node\* prev; // ссылка на предыдущий элемент

struct node\* next;

};

struct node\* head = NULL, \* last = NULL; // указатели на первый и последний элементы списка

void in(const char\* str) {

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

if (last == NULL) {

last = list;

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

last->next = NULL;

}

else {

list->next = last;

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

last = list;

}

}

void out() {

if (last == NULL) {

("Стек пуст\n");

return;

}

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

printf("Извлечённый элемент: %s\n",last->inf);

free(last);

last = NULL;

}

else {

printf("Извлечённый элемент: %s\n", last->inf);

node\* tmp = last;

last = last->next;

free(tmp);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

char data[256];

int buf;

while (1) {

if (last != NULL) {

node\* list = last;

printf("Стек: ");

while (1) {

printf("%s ", list->inf);

if (list->next == NULL) {

printf("\n");

break;

}

list = list->next;

}

}

printf("Введите: 1 для помещения в стек, 2 для извлечения из стека, 0 для выхода ");

scanf("%d", &buf);

printf("\n");

if (buf == 1) {

printf("Введите данные для помещения в стек: ");

scanf("%s", data);

printf("\n");

in(data);

}

else if (buf == 2) {

out();

}

else if (buf == 0) {

exit(0);

}

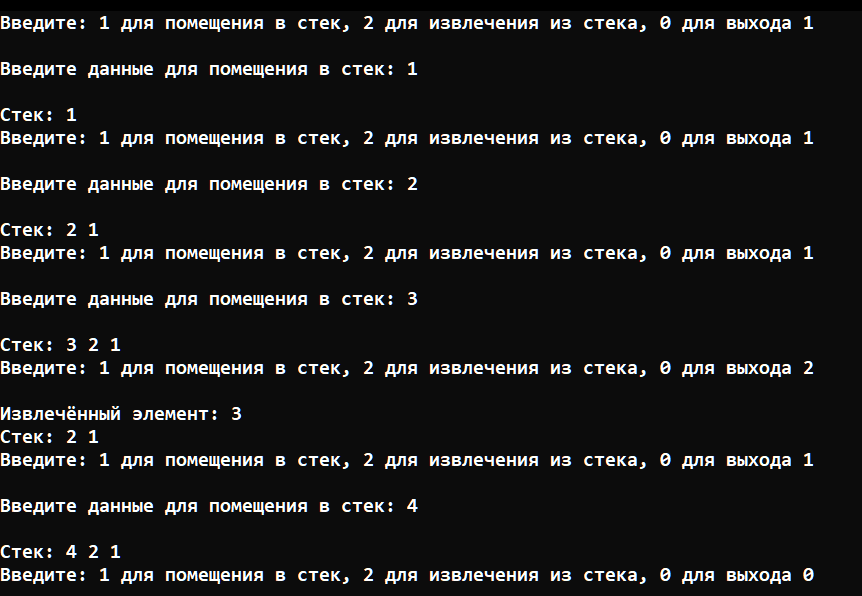
else {

printf("Вы ввели команду неверно\n");

}

}

}

****

**Вывод:** Мы научились реализовывать структуры данных в виде приоритетной очереди, очереди, стек в динамических списках.