Министерство образования РоссийскойФедерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по теме «Динамические списки»

Выполнили:

Студенты группы 21ВВ2

Изосин А.А.

Волдырев В.С.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза, 2022

**Цель работы:**

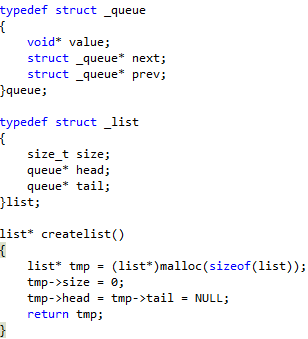
Изучение организации списковых структур и построение реальных структур данных на базе списков.

**Лабораторные работы:**

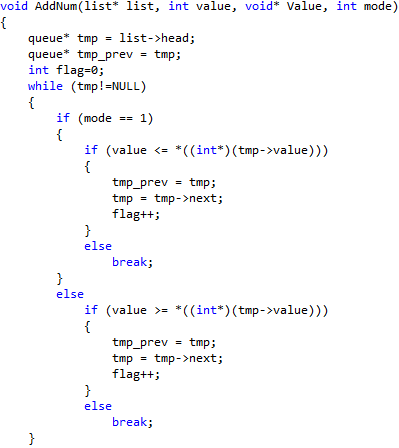
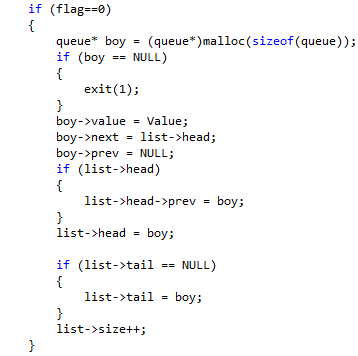
**Задание:**

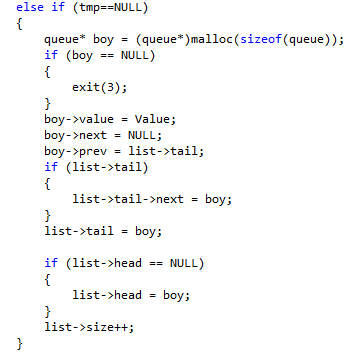
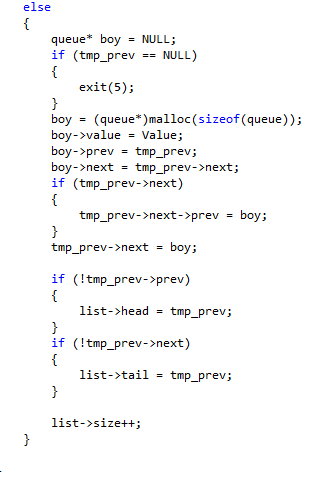
1. Реализовать приоритетную очередь, путём добавления элемента в список в соответствии с приоритетом объекта (т.е. объект  с большим приоритетом становится перед объектом с меньшим приоритетом).
2. \* На основе приведенного кода реализуйте структуру данных *Очередь*.
3. \* На основе приведенного кода реализуйте структуру данных *Стек*.

**Пояснительный текст к программе:**

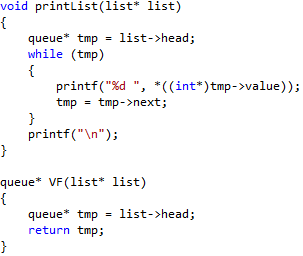
****

В структурах создаем двусвязный список

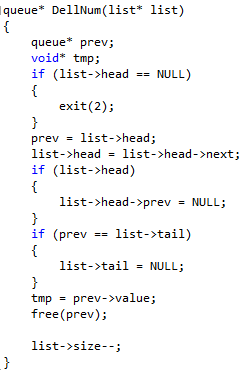


****

Создали функцию добавления чисел в двусвязный список по приоритету, выбираемым пользователем



Добавили функции вывода всего списка и первого элемента списка



Реализовали функцию удаления первого элемента списка

**Листинг:**

#include "stdafx.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

typedef struct \_queue

{

void\* value;

struct \_queue\* next;

struct \_queue\* prev;

}queue;

typedef struct \_list

{

size\_t size;

queue\* head;

queue\* tail;

}list;

list\* createlist()

{

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

tmp->size = 0;

tmp->head = tmp->tail = NULL;

return tmp;

}

void AddNum(list\* list, int value, void\* Value, int mode)

{

queue\* tmp = list->head;

queue\* tmp\_prev = tmp;

int flag=0;

while (tmp!=NULL)

{

if (mode == 1)

{

if (value <= \*((int\*)(tmp->value)))

{

tmp\_prev = tmp;

tmp = tmp->next;

flag++;

}

else

break;

}

else

if (value >= \*((int\*)(tmp->value)))

{

tmp\_prev = tmp;

tmp = tmp->next;

flag++;

}

else

break;

}

if (flag==0)

{

queue\* boy = (queue\*)malloc(sizeof(queue));

if (boy == NULL)

{

exit(1);

}

boy->value = Value;

boy->next = list->head;

boy->prev = NULL;

if (list->head)

{

list->head->prev = boy;

}

list->head = boy;

if (list->tail == NULL)

{

list->tail = boy;

}

list->size++;

}

else if (tmp==NULL)

{

queue\* boy = (queue\*)malloc(sizeof(queue));

if (boy == NULL)

{

exit(3);

}

boy->value = Value;

boy->next = NULL;

boy->prev = list->tail;

if (list->tail)

{

list->tail->next = boy;

}

list->tail = boy;

if (list->head == NULL)

{

list->head = boy;

}

list->size++;

}

else

{

queue\* boy = NULL;

if (tmp\_prev == NULL)

{

exit(5);

}

boy = (queue\*)malloc(sizeof(queue));

boy->value = Value;

boy->prev = tmp\_prev;

boy->next = tmp\_prev->next;

if (tmp\_prev->next)

{

tmp\_prev->next->prev = boy;

}

tmp\_prev->next = boy;

if (!tmp\_prev->prev)

{

list->head = tmp\_prev;

}

if (!tmp\_prev->next)

{

list->tail = tmp\_prev;

}

list->size++;

}

}

void printList(list\* list)

{

queue\* tmp = list->head;

while (tmp)

{

printf("%d ", \*((int\*)tmp->value));

tmp = tmp->next;

}

printf("\n");

}

queue\* VF(list\* list)

{

queue\* tmp = list->head;

return tmp;

}

queue\* DellNum(list\* list)

{

queue\* prev;

void\* tmp;

if (list->head == NULL)

{

exit(2);

}

prev = list->head;

list->head = list->head->next;

if (list->head)

{

list->head->prev = NULL;

}

if (prev == list->tail)

{

list->tail = NULL;

}

tmp = prev->value;

free(prev);

list->size--;

}

int \_tmain()

{

setlocale(0,"");

list\* list = createlist();

int a,b,c,d,e,mode;

printf("Выберете приоритет: \n 1)От большего к меньшему \n 2)От меньшего к большему \n");

scanf("%d",&mode);

printf("Введите значение а: ");

scanf("%d",&a);

AddNum(list,a,&a, mode);

printf("Введите значение b: ");

scanf("%d",&b);

AddNum(list,b,&b, mode);

printf("Введите значение c: ");

scanf("%d",&c);

AddNum(list,c,&c, mode);

printf("Введите значение d: ");

scanf("%d",&d);

AddNum(list,d,&d, mode);

printf("Введите значение e: ");

scanf("%d",&e);

AddNum(list,e,&e, mode);

printList(list);

printf("Удалили первый элемент:", \*((int\*)(DellNum(list))));

printList(list);

printf("Первый элемент списка после удаления:%d\n", \*((int\*)(VF(list))->value));

system("pause");

return 0;

}

**Вывод:**Мы изучили организацию списковых структур и построение реальных структур данных на базе списков.