Министерство образования и науки РФ

Тамбовский государственный технический университет.

Кафедра:

Отчетпо лабораторной работе №2

***«Динамические структуры данных»***

по дисциплине :

«Программированию на языке высокого уровня»

Вариант №2

Выполнил: студент группы   
.

Проверил:.

Тамбов 20

**Постановка задачи**

1. Риализовать двунаправленный кольцевой список.
2. Программа выполняет следующие действия:

* создание нового списка;
* добавление нового элемента в любое место списка;
* удаление указанного элемента из списка;
* полное удаление всего списка;
* нахождение элементов по ключу.

3. Разработать пользовательский интерфейс.

**Граф-схема**

1->2 – начальное состояние списка и вывод меню выбора на экран.

2->3 – создание списка и добавление нового элемента.

2->4 ­– добавление элемента в начало списка.

2->5 – добавление элемента в конец списка.

2->6а – ввод значения элемента, после которого следует осуществить вставку.

6а->6b – поиск заданного значения элемента.

6б->6c – вставка элемента после найденного.

2->7а – ввод значения элемента, перед которым следует осуществить вставку (исключая первый элемент).

7а->7b – поиск заданного значения элемента.

7b->7c – вставка элемента перед найденным.

2->8а – ввод значения элемента, который следует удалить.

8а->8b – поиск заданного значения элемента.

8b->8c – удаление найденного элемента.

2->9 – удаление всего списка.

2->10 – вывод из меню программы.

10->11 – очистка памяти.

**Исходный текст программы**

#include <stdio.h>#include <conio.h>#include <stdlib.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

main()

{

struct spisok{ struct spisok \*next, \*prev;int key;} \*other, \*first=NULL;

int k, count, menuKey,n, index=0;int mes=0;

for (;;)

{

clrscr();

if(mes==1)printf("Error\n");

else if (mes==2)printf("skisok uje sozdan!\n");

else if (mes==3)printf ("element otsutstvyet!\n");

else if (mes==4)printf ("elementov slihkom malo! ( <2 )\n");

printf ("Spisok: ");mes = 0;

if (first==NULL){ printf ("Spisok pust!\n"); }

else

{

other = first;

do

{

printf (" %d",other->key);

other = other->next;

}while (other!=first);

}

printf("1-Sozdat spisok\n"2-Dobavit element v nachalo\n3-Dobavit element v konec \n4-Dobavit element posle zadanogo \n5-Dobavit element pered zadanym \n6-Udalit element \n7-Udalit spisok\n0-Ext\n");

scanf ("%d",&menuKey);

if (!menuKey)break;

switch (menuKey)

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//---------------------------Создать\Добавить -----------------------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 1:{

if (first==NULL)

{

first = (struct spisok\*)malloc(sizeof(struct spisok));

first->next=first->prev=first;

printf ("Perv elem: "); scanf(" %d", &first->key);

} else mes=2; break;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//------------------Добавить элемент в начало------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 2:

{

if (first!=NULL)

{

struct spisok \*third;

third = (struct spisok\*)malloc(sizeof(struct spisok));

printf ("Dobavit v nachalo "); scanf("%d",&third->key);

third->next=first; third->prev=first->prev;

first->prev->next=third;

first->prev=third; first=third;

break;

}

else { mes=1; break;}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//------------------Добавление элем в конец------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 3:

if (first!=NULL)

{

struct spisok \*third;

third = (struct spisok\*)malloc(sizeof(struct spisok));

printf ("Dobav v konec ");scanf("%d",&third->key);

third->next=first;

third->prev=first->prev;

first->prev->next = third;

first->prev=third;

break;

}

else { mes=1; break;}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//-------------------Вставка элемента после указанного------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 4:

if (first!=NULL&&first->next!=first)

{

struct spisok \*third;

printf ("Posle kakogo elementa dobavit elem? ");scanf ("%d", &count);

other = first;

while (other->key!=count)

{

if (other->next==first){mes=3; index =1; break;}

other=other->next;

}

if (index) break;

third=(struct spisok\*)malloc(sizeof(struct spisok));

scanf ("%d",&third->key);

third->prev = other;

third->next=other->next;

other->next->prev = third;

other->next=third;

}

else mes =4; break;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//-------------------Добавить элемент перед указанным-----------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 5:

if (first!=NULL&&first->next!=first)

{

struct spisok \*third;

printf ("Pered kakim dobavit? "); scanf ("%d", &count);

other = first->next;

while (other->key!=count)

{

if (other->next==first){mes=3; index =1; break;}

other=other->next;

}

if (index) break;

third=(struct spisok\*)malloc(sizeof(struct spisok));

scanf ("%d",&third->key);

third->prev = other->prev;

third->next=other;

other->prev->next = third;

other->prev=third;

}

else mes =4; break;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//------------------------------Удалить один элемент списка -------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 6:

if (first==NULL) mes = 1;

else if (first!=NULL&&first->next!=first)

{

printf ("Kakoi element udalit?");scanf ("%d", &count);

other = first;

if (other->key==count)

{

first->prev->next=first->next;

first->next->prev=first->prev;

first=first->next;

free (other); index=1;

}

else if (other->prev->key==count)

{

other=first->prev;

first->prev->prev->next=first;

first->prev=first->prev->prev;

free (other); index = 1;

}

if (index) break;

while (other->key!=count)

{

if (other->next==first){mes=3; break;}

other=other->next;

}

if (other->key==count)

{

other->prev->next=other->next;

other->next->prev=other->prev;

free(other); break;

}

}

else if (first->next==first)

{

printf ("Kakoi element udalit? ");scanf ("%d",&count);

if (first->key==count)

{

free (first);

first=NULL;

}

else mes=3;

break;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//------------------------------Очистка списка----------------------------//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

case 7:

if (first != NULL)

{

while (first->next!=first)

{

other=first;

first->prev->next=first->next;

first->next->prev = first -> prev;

first=first->next;

free (other);

}

free (first);

first = NULL;

break;

}

}

index = 0;

}

if (first != NULL)

{

while (first->next!=first)

{

other=first;

first->prev->next=first->next;

first->next->prev = first -> prev;

first=first->next;

free (other);

}

free (first);

first = NULL;

}

}