

## Operatii cu liste

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct lista
{
    char denumire[20];
    int valoare;
    struct lista *next;
}lista;

lista *head;
```

```
int creare(int n)
{
    int i;
    lista *el,*nod;
    for(i=0;i<n;i++){
        el=(lista*) malloc (sizeof(lista));

        if(el==NULL){
            return 0;
        }
        if(i==0){
            head=el;
            nod=el;
        }
        else{
            nod->next=el;
            nod=el;
        }
    }
    nod->next=NULL;
    return 1;
}
```

```
void read(){
    lista *el=head;
    int i=0;
    while(el != NULL){
        printf("Denumirea: ");
        scanf("%s",el->denumire);
        printf("Introdu valoarea: ");
        scanf("%d",&el->valoare);
        el=el->next;
        i++;
    }
}
```

```

void afisare(){
    lista *el=head;
    int i=0;
    while(el != NULL){
        printf(" %s:",el->denumire);
        printf(" %d\n", el->valoare);
        el=el->next;
        i++;
    }
}

```

```

void cautare_val(){
    lista *el=head;
    int k, count=0, poz=0;

    //cautarea dupa valoare
    puts("\nDati valoarea de cautat\n");
    scanf("%d",&k);
    while(el != NULL){
        poz++;
        if((el->valoare) == k){
            printf("%s: %d a fost gasit pe pozitia %d\n", el->denumire, el->valoare, poz);
            count++;
        }
        el=el->next;
    }
    if(count==0)
        printf("Nu exista asa valoare!");
}

```

```

void cautare_den(){
    lista *el=head;
    int count=0, poz=0;
    char den[20];

    //cautarea dupa denumire
    puts("\nDati denumirea de cautat\n");
    scanf("%s", den);
    while(el != NULL){
        poz++;
        if(strcmp(den,el->denumire)==0){
            printf("%s: %d a fost gasit pe pozitia %d\n", el->denumire, el->valoare, poz);
            count++;
        }
        el=el->next;
    }
    if(count==0)
        printf("Nu exista asa valoare!");
}

```

```

void InserareInceput()
{
    lista *el;
    // Alocare nod si initializare valoare
    el=(lista*) malloc (sizeof(lista));
    printf("Denumirea: ");
    scanf("%s",el->denumire);
    printf("Introdu valoarea: ");
    scanf("%d",&el->valoare);

    // legare nod in lista
    el->next = head;

// mutarea capului listei
head = el;
}

```

```

void InserareSfarsit()
{
    lista *el;
    // Alocare nod si initializare valoare
    el=(lista*) malloc (sizeof(lista));
    printf("Denumirea: ");
    scanf("%s",el->denumire);
    printf("Introdu valoarea: ");
    scanf("%d",&el->valoare);

    // daca avem lista vida
    if (head == NULL)
        // doar modificam capul listei
        head = el;
    else
    {
        // parcurgem lista pana la ultimul nod
        lista *nod = head;
        while (nod->next != NULL)
            nod = nod->next;
        // adaugam elementul nou in lista
        nod->next = el;
    }
}

```

```

void InserareInterior()
{
    lista *el;
    int poz=0, k;
    // Alocare nod si initializare valoare
    el=(lista*) malloc (sizeof(lista));
    printf("Denumirea: ");
    scanf("%s",el->denumire);
    printf("Introdu valoarea: ");
    scanf("%d",&el->valoare);

    // lista vida
    if (head == NULL)
    {
        head = el;
        return;
    }
    puts("\nDati pozitia dupa care element doriti inserarea el. nou \n");
    scanf("%d",&k);
    lista *nod = head;
    while (nod->next != NULL){
        poz++;
        if(poz == k){
            // inserare in interior
            el->next = nod->next;
            nod->next = el;
        }
        nod=nod->next;
    }
}

```

```

void stergeElement(){
    int poz, k=0;
    // daca lista e vida nu facem nimic
    if (head == NULL)
        return;
    // daca este primul element, atunci
    // il stergem si mutam capul
    puts("\nIntroduceti pozitia elementului de exclus \n");
    scanf("%d",&poz);
    if (poz == 1)
    {
        lista* deSters = head;
        head = head->next;
        free(deSters);
        return;
    }
    //sterge element din interior
    lista *nod = head;
    lista* deSters;
    while (nod->next != NULL){
        k++;
        if(poz == k){
            deSters = nod->next;
            nod->next = nod->next->next;
            free(deSters);
        }
        nod = nod->next;
    }
}

```

```

int main()
{
    int n;
    printf("Introdu numarul de elemente din lista\n");
    scanf("%d",&n);
    creare(n);
    read();
    afisare();
    cautare_val();
    cautare_den();
    InserareInceput();
    InserareSfarsit();
    InserareInterior();
    stergeElement();
    afisare();

    return 0;
}

```