

Liste simplu înlănțuite

https://github.com/alexchirea/TutoriatPC2018/blob/master/liste_simplu_inlantuite.c

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

/* Structura unui nod */
typedef struct nod {
    int info;
    struct nod *next;
} nod;

/* Functie inserare nod nou la sfarsitul listei */
void inserare(int info, nod **st, nod **sf)
{
    nod *noulNod = (nod*)malloc(sizeof(nod));
    noulNod->info = info;
    noulNod->next = NULL;
    if (*st == NULL)
    {
        *st = noulNod;
        *sf = noulNod;
    }
    else
    {
        (*sf)->next = noulNod;
        *sf = noulNod;
    }
    return;
}

/* Functie parcurgere si afisare lista */
void afis(nod *st)
{
    nod *aux = st;
    while (aux != NULL)
    {
        printf("%d ", aux->info);
        aux = aux->next;
    }
    printf("\n");
    return;
}

/* Functie care cauta o valoare si intoarce adresa nodului */
nod* cautare(int deCautat, nod *st)
{
    nod *aux = st;
    while (aux != NULL)
    {
        if (aux->info == deCautat)
            return aux;
        aux = aux->next;
    }
    return NULL;
}

/* Functie de stergere a unui nod ce contine valoarea x */
void stergere(int x, nod **st, nod **sf)
```

```

{
    nod *aux = *st, *anterior;
    while (aux != 0)
    {
        if (aux->info == x)
        {
            if (aux == *st) /* Caz I: primul nod */
            {
                *st = aux->next;
                free(aux);
                aux = *st;
            }
            else if (aux == *sf) /* Caz II: ultimul nod */
            {
                free(aux);
                *sf = anterior;
                anterior->next = 0;
            }
            else /* Caz III: in interioriul listei */
            {
                anterior->next = aux->next;
                free(aux);
                aux = anterior;
            }
        }
        anterior = aux;
        aux = aux->next;
    }
    return;
}

int main()
{
    nod *start, *sfarsit;
    start = sfarsit = 0; /// 0 == NULL
    int n, i, x;
    scanf("%d", &n);
    for(i = 0; i < n; i++)
    {
        scanf("%d", &x);
        inserare(x, &start, &sfarsit);
    }
    afis(start);
    stergere(5, &start, &sfarsit);
    afis(start);
    stergere(1, &start, &sfarsit);
    afis(start);
    stergere(8, &start, &sfarsit);
    afis(start);

    /**
     * EXEMPLU
     *
     * stdin:
     * 10 1 3 4 5 9 6 5 8 1 8
     *
     * stdout:
     * 1 3 4 5 9 6 5 8 1 8
     * 1 3 4 9 6 8 1 8
     * 3 4 9 6 8 8
     * 3 4 9 6

```

```
    ** /  
}
```