// Sa se scrie o procedura care primeste o lista de numere intregi. Procedura muta elementele acestei liste in alte 2 liste (initial vide) astfel:

nr 1: - elementele negative in prima lista

- elementele pozitive in a doua lista

nr 2: - elementele impare in prima lista

- elementele pare in a doua lista

#include <stdio.h>

#include <stdlib.h>

typedef struct str\_l{int info;

struct str\_l \*urmator;

} element;

int introducere(element \*\*lista, int info)

{element \*t;

t=(element\*)malloc(sizeof(element));

if(t==NULL)return (1);

t->info=info;

t->urmator=\*lista;

\*lista=t;

return(0);

}

void afisare(element \*lista)

{element \*p;

p=lista;

while(p!=NULL)

{printf("%d ",p->info);

p=p->urmator;

}

}

int extragere(element \*\*lista, int \*var)

{element \*t;

t=\*lista;

if(t==NULL)

return(1);

if(t->urmator==NULL)

\*lista=NULL;

else (\* lista= t->urmator);

free(t);

return(0);

}

int main()

{element \*lista, \*l1, \*l2;

int r,x;

lista=NULL;

l1=NULL;

l2=NULL;

printf("doriti sa introduceti un nr? 0=DA si 1=NU");

scanf("%d",&r);

if (r!=1)

do{printf("introduceti un nr:");

scanf("%d",&x);

introducere(&lista, x);

printf("mai doriti sa introd? 0=DA 1=NU");

scanf("%d",&r);

}while(r!=1);

if(lista==NULL)

printf("lista vida");

while(extragere(&lista, &x)!=1)

if(x%2==0)introducere(&l1, x);

else introducere(&l2, x);

printf("nr pare: ");

afisare(l1);

printf("nr impare: ");

afisare(l2); }

// Structura de date asociata unui element al listei

typedef struct str\_element{ int info;

struct str\_element \*urmator;

}element;

element \*lista; // Indicator catre primul element din lista

// Procedura de afisare a elementelor din lista

void afisare\_lista(void)

{

element \*p;

if(lista==NULL){

printf("Lista este vida\n");

return;

}

printf("Lista: ");

p=lista;

while(p!=NULL){

printf("%d ",p->info);

p=p->urmator;

}