

## Cerință

Având la dispoziție o listă de date dublu înlănțuită—i.e., listă în care fiecare element are o legătură către nodul următor, dar și către nodul anterior, creată prin intermediul bibliotecii atașate, să se implementeze următoarele funcționalități:

- (1) o funcție care să permită afișarea datelor din fiecare nod din listă, în ordine inversă, începând cu ultimul element introdus;
- (2) o funcție care să permită inserția unui nod pe prima poziție în lista de date.
- (**Bonus**) Să se declare și apelze o funcție care să permită ștergerea și eliberarea memoriei fiecărui nod din lista dublu înlănțuită.

## Date de intrare

Se vor citi de la tastatură (fluxul *stdin*) următoarele date:

- O valoare întreagă  $n$  reprezentând numărul de noduri din lista de date, urmată de tasta Enter;
- Pentru fiecare nod în parte, un număr întreg reprezentând valoarea stocată în membrul *date*, urmat de tasta Enter.
- O valoare întreagă  $x$  reprezentând valoarea stocată în nodul care va fi inserat pe prima poziție din lista de date, urmată de tasta Enter;

## Date de ieșire

Programul va afișa pe ecran la ieșire:

- Valoarea din membrul *date* al fiecărui nod al listei de date dublu înlănțuite, fiecare pe o linie nouă, în ordinea inversă față de cea în care au fost citite de la tastatură (începând de la ultimul nod, către primul);
- Valoarea din membrul *date* al fiecărui nod al listei de date dublu înlănțuite, fiecare pe o linie nouă, în ordinea inversă față de cea în care au fost citite de la tastatură, după inserția unui nod pe prima poziție a listei de date.

**ATENȚIE la respectarea cerinței problemei: afișarea rezultatelor trebuie făcută EXACT în modul în care a fost indicat! Cu alte cuvinte, pe stream-ul standard de ieșire nu se va afișa nimic în plus față de cerința problemei; ca urmare a evaluării automate, orice caracter suplimentar afișat, sau o afișare diferită de cea indicată, duc la un rezultat eronat.**

## Precizări

- Biblioteca atașată permite declararea și inițializarea unei liste de date dublu înlănțuită cu  $n$  noduri ale căror date sunt introduse de la tastatură. Nodurile relaționează între ele prin intermediul pointerilor *struct LISTA \*anterior*, *struct LISTA \*urmator*, care stochează adresele de memorie a nodului anterior și respectiv, următor din lista de date;
- Funcția *struct LISTA \*inserareNod()* este o funcție generală, prin intermediul căreia se creează și se alocă memorie pentru un nod nou. Conținutul acestuia este inițializat cu informația citită de la tastatură, iar locația nodului următor este inițializată cu *NULL*, deoarece în prima instanță acest nod nu este inserat în listă. Dacă lista este nulă, atunci noul nod devine primul nod din listă iar adresa lui este returnată printr-un pointer de tip

*struct LISTA* altfel, noul nod este adăugat la sfârșitul listei de date prin parcurgerea listei cu ajutorul structurii repetitive *while*, cu condiția de oprire activată de nodul a cărui adresă este *NULL* prin expresia: *nodCurent!=NULL*.

## Exemplu

Intrare	Ieșire
4	9
2	5
7	7
5	2
9	9
-1	5
	7
	2
	-1

**Timp de lucru: 60 de minute**