

Cerință

Având la dispoziție o listă de date circulară simplu înlănțuită—i.e., listă al cărui ultim nod introdus nu mai indică spre adresa *NULL*, ci către primul nod introdus în listă, creată prin intermediul bibliotecii atașate, să se implementeze următoarele funcționalități:

- (1) o funcție care să permită afișarea valorii din membrul *date* al fiecărui nod al listei de date;
 - (2) o funcție care să insereze un nod înaintea primului nod dintr-o listă de date circulară simplu înlănțuită, iar acesta să devină nodul care să permită intrarea în lista de date circulară existentă.
- (**Bonus**) Să se declare și apeleze o funcție care să permită atât eliminarea, cât și eliberarea memoriei a fiecărui nod dintr-o listă de date circulară simplu înlănțuită în ordinea în care au fost citite de la tastatură.

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 reprezentând valoarea membrului *date* a nodului introdus înaintea primului nod dintr-o listă de date circulară simplu înlănțuită existentă, urmată de tasta Enter (vezi (**Bonus**)).

Date de ieșire

Programul va afișa pe ecran la ieșire:

- Valoarea din membrul *date* al fiecărui nod al listei de date circulară simplu înlănțuită, fiecare pe o linie nouă, în ordinea în care au fost citite de la tastatură;
- Valoarea din membrul *date* al fiecărui nod al listei de date circulare simplu înlănțuită după adăugarea unui nod înaintea primului nod dintr-o listă de date circulară simplu înlănțuită existentă, fiecare pe o linie nouă.

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 circulară simplu înlănțuită cu *n* noduri ale căror date sunt introduse de la tastatură. Nodurile relaționează între ele prin intermediul pointerului *struct LISTA *urmator*, care stochează adresa de memorie a nodului 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 lista de date. 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 legătură indică spre primul nod introdus în lista de date prin expresia: *nodCurent->următor != lista*.

Restricții

1. Interschimbarea nodurilor se va realiza prin refacerea legăturilor atât a nodurile interschimbate precum și vecinilor lor.

Exemplu

Intrare	Ieșire
5	4
4	9
9	15
15	2
2	1
1	-1
-1	4
	9
	15
	2
	1

Timp de lucru: 60 de minute