PROBLEME - LISTE

1. Într-o bibliotecă sunt mai multe teancuri de cărţi. Bibliotecarul vrea să reţină doar cărţile având anul de apariţie mai mare decât 1970, dar în ordine alfabetică a titlurilor. Să se afişeze cărţile în ordinea în care trebuie reţinute pe raft. **<u>Indicaţie:</u> Stivă, CoadăCuPriorităţi**.

2. **<u>Jocul Gâsca Roşie</u>**. 2 jucători primesc iniţial $\frac{n}{2}$ cărţi de joc (fiecare carte poate avea culoarea roşie sau neagră). Jucătorii pun alternativ câte o carte pe masă (din vârful teancului lor de cărţi), până se pune o carte roşie (caz în care teancul de pe masă va fi luat de către jucătorul care nu a pus cartea roşie şi adăugat sub teancul său de cărţi). Pierde jucătorul care nu mai are cărţi. Simulaţi jocul. **<u>Indicaţie:</u> Stivă, Coadă**.

3. **<u>Problema lui Joseph</u>**. n persoane se află în cerc: persoanele se elimină din m în m, începând cu persoana cu numărul k. Se cere să se afişeze ordinea în care vor fi eliminate persoanele din cerc. **<u>Indicaţie:</u> Listă circulară**.

4. Fie un labirint (reţea dreptunghiulară) cu celule ocupate (X) şi libere (∗). Fie R un robot în acest labirint. Robotul se poate deplasa în 4 direcţii: N, S, E, V.

∗	∗	X	∗	∗
X	∗	∗	∗	X
X	∗	R	∗	∗
X	∗	X	X	∗
∗	∗	∗	∗	∗

a). Testaţi dacă R poate ieşi din labirint (poate ajunge la margine).
b). Determinaţi un drum pentru ieşire (dacă există).

<u>Indicaţie</u>

Fie T mulţimea poziţiilor în care robotul poate ajunge pornind de la poziţia iniţială. Notăm cu S mulţimea poziţiilor în care robotul a ajuns până la un moment dat şi din care s-ar putea deplasa. Un algoritm pentru determinarea mulţimilor T şi S ar putea fi:

T←{poziţia iniţială}
S←{poziţia iniţială}
Cât-timp $S \neq \phi$ execută
    Fie p un element din S
    $S \leftarrow S \setminus \{p\}$
    Pentru fiecare poziţie q alăturată poziţiei p, $q \neq$'X' şi $q \notin T$ execută
        $S \leftarrow S \cup \{q\}$

$$T \leftarrow T \cup \{q\}$$

SfPentru

SfCatTimp

Observaţii

- Pentru a răspunde la punctul a), algoritmul s-ar putea termina dacă poziţia q care satisface condiţiile este pe frontiera labirintului.
- Mulţimea T poate fi memorată printr-o matrice asociată labirintului (ex: 0 pentru poziţiile neatinse încă, respectiv 1 pentru poziţiile în care robotul a ajuns).
- Structura S poate fi o Stivă sau o Coadă.
- Pentru a răspunde la punctul b), ne putem gândi la un algoritm care pornind de la o poziţie de pe frontieră (în cazul în care răspunsul ala punctul a) a fost pozitiv) merge din aproape în aproape pe poziţiile marcate cu 1 (atinse) spre poziţia iniţială.