

# Programarea Calculatoarelor – Tema 1

## TASK 1

Gigi, student in anul I la Calculatoare, este incepator in programarea in C, insa reuseste sa se acomodeze. Dupa laboratorul 3, ajunge acasa si se apuca sa exerseze instructiunile repetitive (for, while, do while).

Intre timp, Gigi devine din ce in ce mai bun la programare. Fiind un mare pasionat de jocuri, decide sa incerce sa faca un mic joculet in consola (top-down single-player turn-based MMORPG).

Incepe cu ceva simplu: sa creeze un personaj pe care sa-l poata muta folosind tastatura.

O „tabla de joc” este o zona de dimensiune NxM care va fi afisata in consola astfel:

```
*****
*      *
*      *
*      *
*****
```

Acesta e un exemplu de tabla 2×4 (zona libera din interior are 2 randuri și 4 coloane). Pe o tabla va exista un „personaj”, care este reprezentat de caracterul ‘v’.

Jocul este turn-based, iar o tura functioneaza in felul urmator:

- se afiseaza tabla de joc curenta;
- se citeste de la tastatura un singur caracter de miscare (w, a, s sau d) urmat de caracterul newline (\n);
- personajul se muta pe tabla corespunzator (w=sus, a=stanga, s=jos, d=dreapta);
- se trece la tura următoare.

Jocul se considera incheiat cand personajul ajunge in coltul din dreapta jos al tablei, caz in care se afiseaza tabla corespunzatoare (dupa aplicarea mutarii ce trimite personajul în coltul din dreapta jos), urmata de mesajul “GAME COMPLETED” pe linia următoare.

Exemplu:

Fie tabla de joc de forma urmatoare, cu pozitia personajului la coordonatele (y,x) = (1,2) (randul 1, coloana 2):

```
*****
* v   *
*     *
*     *
*****
```

Utilizatorul scrie la tastatura caracterul ‘s’, apoi tasteaza Enter dupa aplicarea operatiilor corespunzatoare, programul afiseaza tabla de joc aferenta turei urmatoare (unde personajul a fost “mutat” cu o linie mai jos):

```
*****
*     *
* v   *
*****
```

Daca acum utilizatorul scrie la tastatură caracterul ‘d’ si tasteaza Enter, tabla va deveni:

```
*****
*     *
* >  *
*****
```

(personajul a fost mutat la dreapta)

Dupa inca o mutare la dreapta, personajul ajunge in colt iar jocul se termina. Se afiseaza tabla si mesajul de terminare:

```

*****
*      *
*      >*
*****

```

GAME COMPLETED

Scrieti un program care citeste de la tastatură 5 numere: N, M, x, y, P (din care N, M, x si y cu semnificația data mai sus, iar P va fi momentan 0 – il veti folosi la taskul urmator), afiseaza tabla curenta si aplica miscarea conform cu datele citite de la tastatura. Dupa fiecare miscare citita de la tastatura, tabla de joc trebuie afisata din nou, avand modificarea aplicata.

Format intrare:

Pe prima linie se dau numerele N, M, x, y, 0 separate prin spatiu.

Exemplu:

2 4 2 1 0

Pe fiecare linie care urmeaza se va citi un caracter de miscare (w/a/s/d) sau caracterul q.

Programul se incheie la citirea caracterului q (fara sa mai afiseze inca o data tabla).

Exemplu iesire:

Dupa citirea primei linii (cu numerele N, M, x, y, P) si dupa fiecare caracter de miscare citit de la tastatura, tabla de joc curenta trebuie afisata.

```

*****
*  v  *
*      *
*****

```

Observatii și restrictii:

- $1 \leq y \leq N \leq 80$
- $1 \leq x \leq M \leq 80$
- coordonatele primei pozitii libere din stanga-sus sunt (1,1);
- daca personajul se afla intr-una din margini (delimitate de asteriscuri) iar miscarea ceruta de utilizator este inspre marginea respectiva, miscarea personajului nu trebuie sa se efectueze (tabla de joc pentru runda urmatoare va fi identică cu cea de la runda precedenta) la afisarea unei table de joc, după ultima linie de asteriscuri trebuie afișat un singur caracter newline (\n);
- caracterul cu care este afisat personajul se modifica în functie de directia pe care personajul a fost mutat ultima oara. Se vor folosi caracterele 'v', '^', '<', '>' pentru jos, sus, stanga si, respectiv, dreapta;
- la inceput (dupa citirea primei linii de la tastatură), personajul va fi implicit orientat in jos si va folosi caracterul 'v' pentru afisare.

## TASK 2

Avand ambitii de gamer profesionist, lui Gigi i se pare ca jocul este prea simplu, astfel incat decide sa-i adauge niste obstacole.

Numărul P citit de la tastatură pe prima linie reprezintă numărul de obstacole. Dacă  $P > 0$ , pe următoarele P linii se află informațiile despre fiecare obstacol: câte două numere l și c, și un caracter m, cu semnificația de mai jos.

- l – linia pe care se află obstacolul;
- c – coloana pe care se află obstacolul;
- m – un caracter ce reprezintă modul de funcționare al obstacolului: 'f' pentru obstacol fix, și 'm' pentru obstacol mobil.

Un obstacol va fi reprezentat pe tabla de joc folosind caracterul 'x'. Un obstacol fix va ramane mereu acolo unde a fost positionat, la coordonatele (l, c). Un obstacol mobil se va deplasa inspre personaj, pe una din cele 8 directii, cu 1 pozitie la fiecare 2 ture. Cand utilizatorul executa o mutare care trimite personajul intr-o pozitie ocupată de un obstacol (sau cand un obstacol mobil vine peste personaj), jocul se termina (nu se mai afiseaza tabla de joc inca o data, in schimb se afiseaza mesajul "GAME OVER" si executia se incheie).

Modificati programul de pana acum ca sa ia in considerare varianta in care  $P > 0$ .

Observatii si restrictii:  $0 \leq P \leq 3$

Exemplu intrare

3 4 2 1 3

2 1 f

1 4 m

3 4 m

Semnificatie: Tabla este 3×4, personajul se afla la coordonatele (1,2) si exista 3 obstacole: unul la coordonatele (2,1) fix, unul mobil la coordonatele (1, 4), unul mobil la coordonatele (3, 4).  
Tabla de joc va arăta astfel:

```
*****
*  V  X*
*X      *
*      X*
*****
```

Daca utilizatorul executa doua mutari succesive in jos, dupa prima mutare tabla este:

```
*****
*      X*
*XV    *
*      X*
*****
```

Dupa a doua mutare (au trecut doua ture, deci obstacolele mobile se vor muta cu o pozitie spre personaj):

```
*****
*      *
*X X  *
* VX  *
*****
```

Dupa o noua mutare la dreapta, nu se mai afiseaza tabla de joc inca o data, ci se afiseaza urmatorul mesaj si programul se incheie:  
GAME OVER

Observatii finale:

- toate coordonatele (pentru personaj si obstacole) vor avea valori valide (se vor afla in interiorul tablei de joc, nu vor avea valori negative, nu se vor suprapune etc.), deci nu e nevoie de verificari suplimentare;
- dupa oricare din mesajele “GAME OVER” sau “GAME COMPLETED” trebuie afisat caracterul newline (\n)
- **nu este permis să folositi tablouri uni/multi-dimensionale pentru rezolvarea cerintelor temei sau variabile globale**

Un obstacol mobil se va deplasa inspre personaj alegand pozitia adiacenta care minimizeaza functia:  $|x_{\text{personaj}} - x_{\text{obstacol}}| + |y_{\text{personaj}} - y_{\text{obstacol}}|$   
Obstacolele pot ajunge deci sa se suprapună de-a lungul jocului.