

**Proiect Seminar 2023-2024**

# **— SNAKES AND LADDERS —**

Spătărelu  
Eduard  
~  
Sticlan  
Daiana

# Cuprins



## 1. Introducere

- 1.1 Enunțul problemei
- 1.2 Descrierea problemei

## 2. Algoritmi

- 2.1 Pseudocod
- 2.2 Scheme logice



## 3. Descrierea aplicației

- 3.1 Utilizare
- 3.2 Avantaje
- 3.3 Dezavantaje

## 4. Rezultate

## 5. Concluzii

## 6. Appendix

## 7. References



01

# Introdúcere

# 1.1 Enunțul problemei

Snakes and Ladders este cel mai frecvent joc de societate jucat. Regulile jocului sunt următoarele:

- \*Prima persoană care ajunge la 100 câștigă.
- \*Fiecare jucător are o singură șansă într-o singură traversare.
- \*Șerpii îți scad punctele în timp ce scara le mărește.

Deci, deoarece regulile sunt destul de ușor de înțeles, le putem codifica cu ușurință folosind limbajul C pentru a crea o aplicație Snakes and Ladders. Funcționalitatea codului va fi după cum urmează:

- \*Doi jucători pot intra într-un singur joc.
- \*Valorile aleatorii pot fi obținute folosind zaruri pentru a crește sau a reduce valoarea.
- \*Punctele vor fi menținute folosind variabile.
- \*Jocul se va încheia după ce orice jucător atinge 100 de puncte.

## 1.2 Descrierea problemei

1

Punctele reținute în variabile reprezintă de fapt poziții pe tabla de Snakes and Ladders.

2

Fiecare jucător începe de pe poziția 1, prima poziție de pe tablă, avansând în funcție de valoarea indicată de zar.

3

Aruncarea zarului se face succesiv de către cei doi participanți.

4

Când poziția unui jucător corespunde unei scări, acesta va avansa până la capătul scării (valoarea poziției va crește).

5

Când poziția unui jucător corespunde capătului unui șarpe, acesta se va întoarce pe tablă câte poziții îi indică șarpele.

# Template-ul utilizat

Conform tablei de joc aleasă:

❖ Pozițiile de început și de final ale SCĂRILOR sunt: 2-23, 8-34, 20-77, 32-68, 41-79, 74-88, 82-100, 85-95;

❖ Pozițiile de final și de început ale ȘERPILOR sunt: 29-9, 38-15, 47-5, 53-33, 62-37, 86-54, 92-70, 97-25;

❖ Primele numerele din fiecare pereche de la punctele 1) și 2) ale acestei secțiuni reprezintă poziții critice care măresc sau scad punctajul.



02

# Algoritmi



## 2.1 Pseudocod

### START PROGRAM

Initialize random number generator  
Open input and output files  
IF error opening files THEN  
    Print error message and exit program  
END IF

Write game start message to output file  
Call game function with input and output  
Close input and output files

### END PROGRAM

### FUNCTION Throw Biased Dice(bias)

Generate random number between 0 and 1  
IF random number is less than bias THEN  
    Return random number between 4 and 6

ELSE

    Return random number between 1 and 3

END IF

### FUNCTION Calculate Bias(position)

IF position is less than 50 THEN

    Return 0.5

ELSE

    Return number that linearly increases from 0.5 to 0.9 as position increases from 50 to 100

END IF

END FUNCTION

You can replace this picture if you wish

### FUNCTION Check for Snakes and Ladders(position)


Depending on position, return certain number representing a snake or ladder

case 2: return 21  
case 8: return 26  
case 20: return 57  
case 32: return 36  
case 41: return 38  
case 74: return 14  
case 82: return 18  
case 29: return -20  
case 38: return -23

case 47: return -42  
case 53: return -20  
case 62: return -25  
case 86: return -32  
case 92: return -22  
case 97: return -72  
default: return 0

END FUNCTION





### FUNCTION Play Round(player number, player position, input file, output file)

Call Wait for 'z' Press function with input and output files  
Calculate bias using Calculate Bias function  
Throw biased dice using Throw Biased Dice function  
Update player position  
Check for snakes or ladders using Check for Snakes and Ladders function  
IF there's a snake or ladder THEN

Update player position and print appropriate message

END IF

IF player position is greater than 100 THEN

Set player position to 100

END IF

Print player's current position

Return player position

END FUNCTION



### FUNCTION Play Game(input file, output file)

Initialize positions of player 1 and player 2

WHILE neither player has reached position 100 DO

Update position of player 1 using Play Round function

IF player 1 has reached position 100 THEN

Print victory message and break loop

END IF

Update position of player 2 using Play Round function

IF player 2 has reached position 100 THEN

Print victory message and break loop

END IF

END WHILE

END FUNCTION

### FUNCTION Read Z

Read character from input file WHILE character is not 'z' DO

IF character is EOF THEN

Print error message and exit program

ELSE IF character is not a newline THEN

Print message asking for 'z'

END IF

Read next character from input file

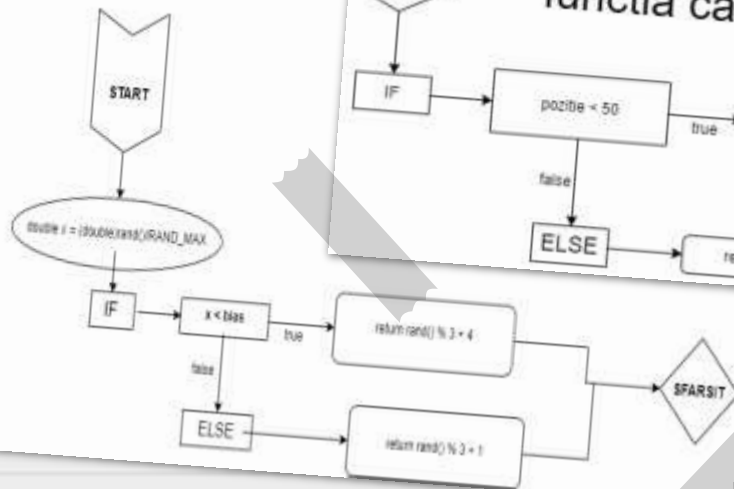
END WHILE

END FUNCTION

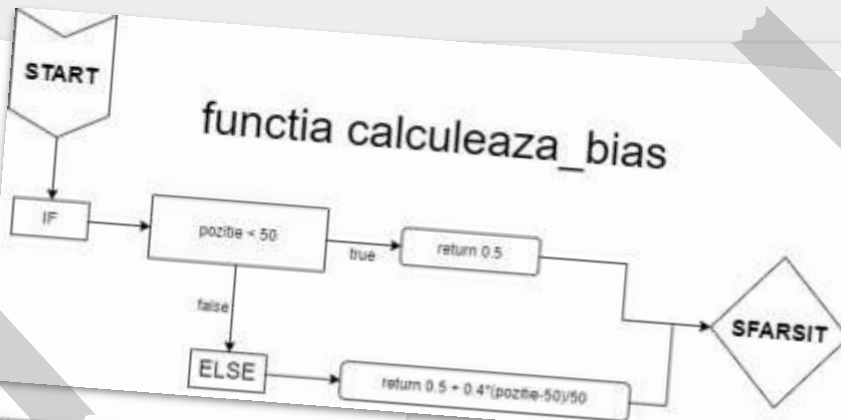


## 2.2 Scheme logice

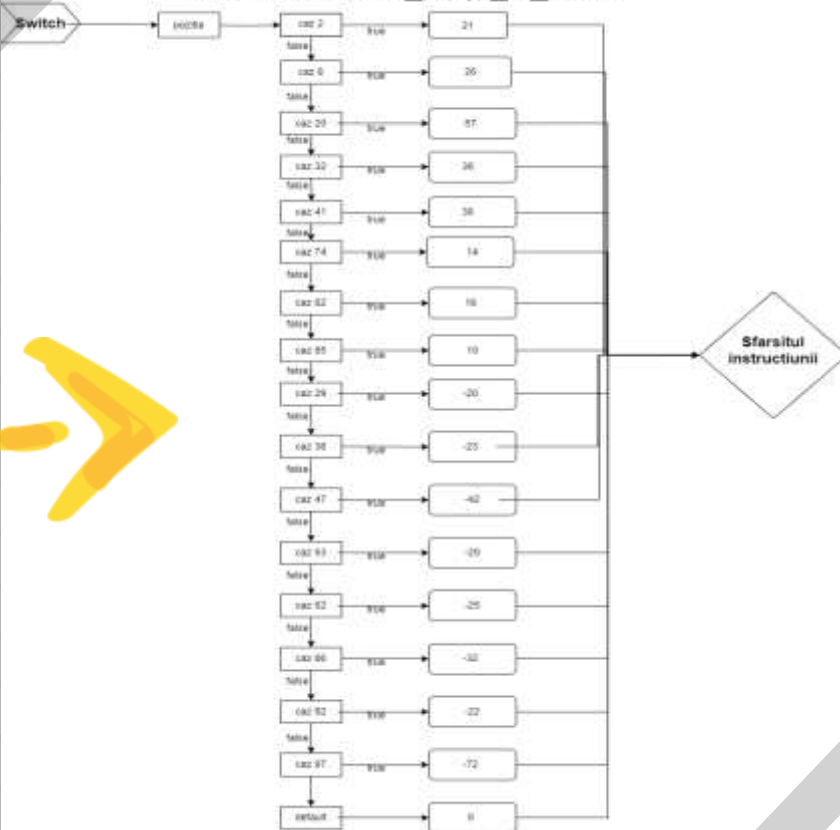
functia arunca\_zarul\_biasat



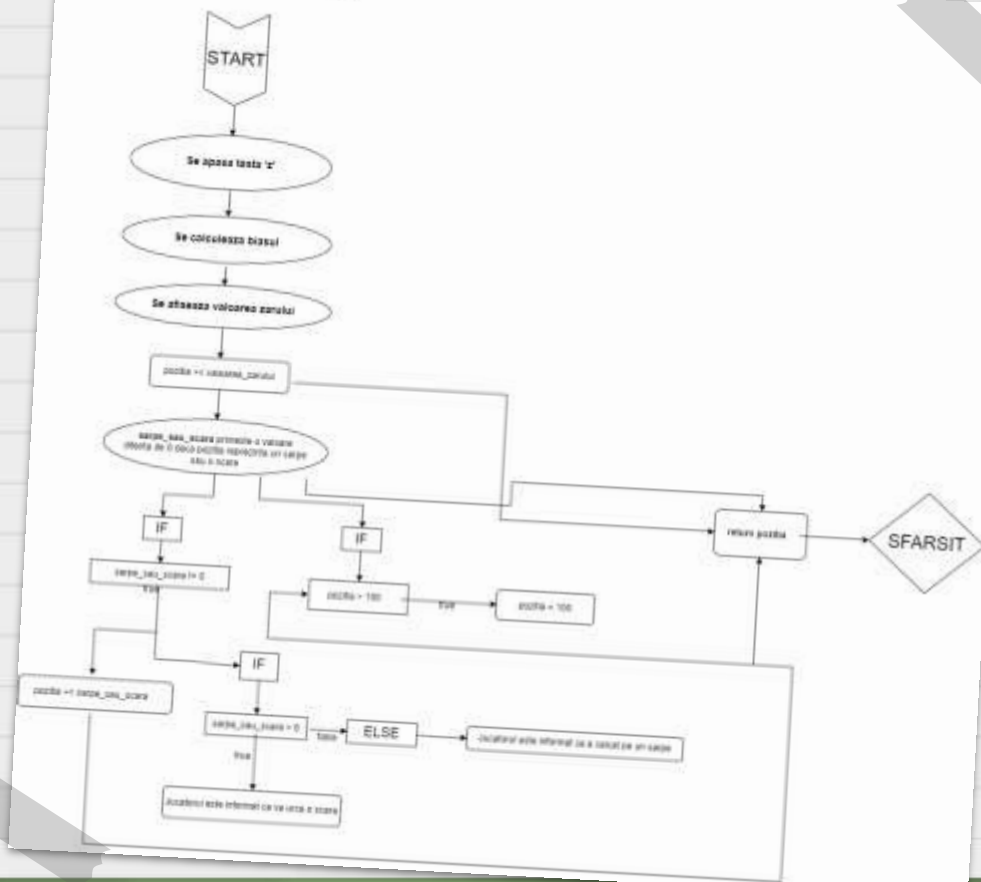
functia calculeaza\_bias



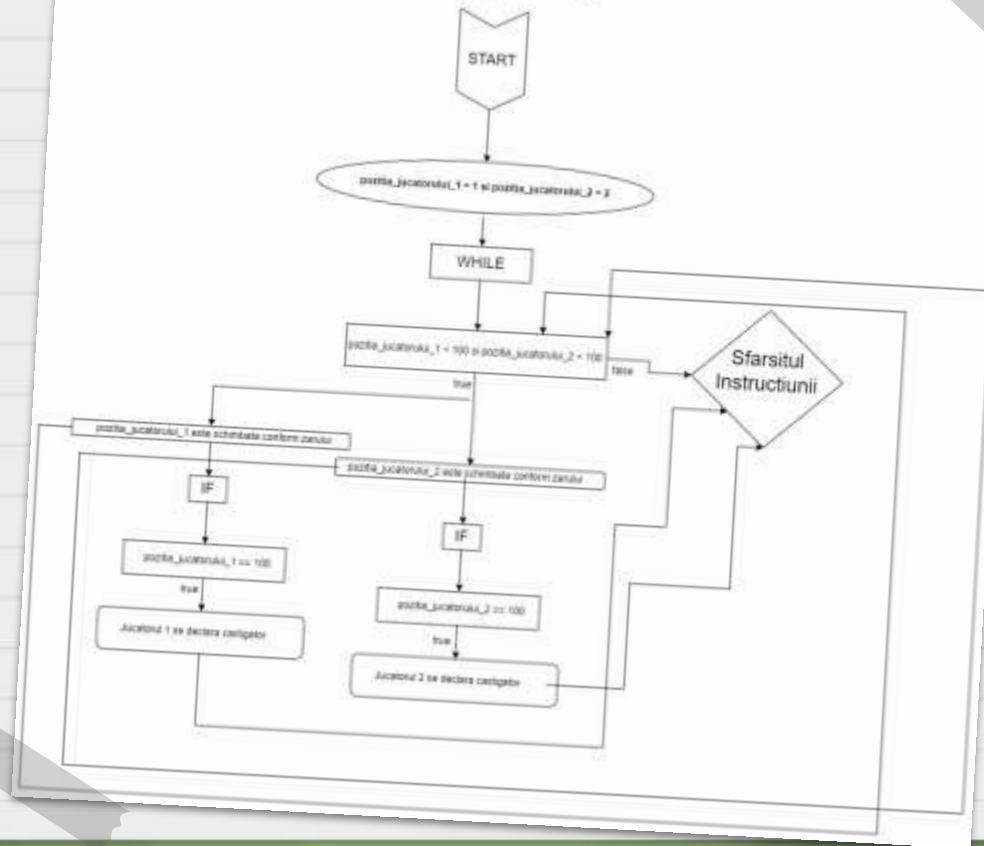
## Funcția cautam\_serpi\_si\_scari



# functia runda\_noua



## functia sa\_jucam\_jocul





03

## **Descrierea aplicației**

## 3.1 Utilizare

Aplicația are ca principal scop divertismentul și poate ține cu ușurință locul jocului clasic de Snakes and Ladders. Punctajul care reprezintă poziția fiecărui dintre cei doi jucători este reținut cu ajutorul variabilelor.

De fiecare dată când un jucător întâlnește o scară (corespunzătoare șablonului ales), variabila `pozitia` este incrementată, deci jucătorul înaintează un număr de pași suplimentari pe tablă. În cazul în care acesta se află într-un cadran corespunzător unui șarpe, valoarea variabilei `pozitia` scade, deci jucătorul se îndepărtează de obiectivul final de a atinge cele 100 de puncte. Incrementările și decrementările se realizează prin intermediul unei instrucțiuni `switch`.

Programul îi semnalează jucătorului dacă are de a face cu un șarpe sau cu o scară afișând un mesaj corespunzător.

În cazul în care un jucător nu apasă tasta 'z' pentru a arunca zarul, există o funcție care sesizează acest lucru și se va crea o buclă până la primirea tastei 'z', în tot acest timp utilizatorul primind mesajul de a apăsa tasta corespunzătoare.





Despre felul în  
care biasul este  
utilizat

Pentru a evita o posibilă durată prea mare a jocului, generarea numerelor care reprezintă fețele zarului s-a realizat în funcție de bias, care este setat la 0.5 (distribuție uniformă) atunci când poziția jucătorului este mai mică decât 50, adică el se află în prima jumătate a tablei și există o șansă egală de a genera un număr între 1 și 3 sau între 4 și 6. Pe măsură ce jucătorul avansează în joc (poziție mai mare sau egală cu 50), biasul crește și totodată crește și probabilitatea de a genera un număr mai mare (între 4 și 6). Acest lucru ajută la scurtarea timpului de joc deoarece, pe măsură ce un jucător se apropie de sfârșitul jocului, zarul său va indica valori mai mari.



## 3.2 Avantaje

Aplicația este foarte ușor de utilizat. Prin apăsarea tastei z, fiecărui jucător îi este atribuită o valoare de la 1 la 6.

Poziția pe tablă este calculată instantaneu, iar programul semnalează situațiile în care apar șerpi sau scări.

Utilizarea a funcției adecvate pentru fiecare acțiune ușurează citirea și întreținerea codului.



## 3.3 Dezavantaje

Jocul are la bază generarea de numere random, similară cu aruncarea zarului din viața reală, fapt care ar fi putut determina o durată lungă a jocului, în funcție de norocul jucătorilor. Din acest motiv, a fost introdusă generarea de numere convenabile, fapt care nu afectează performanța în sine a aplicației, dar care se îndepărtează de modelul real, în care o durată mult prea lungă determina un factor în plus de divertisment.



04

Rezultate



# Prima testare

În următoarele print screen-uri realizate fisierului output, este prezentată rularea programului.

În acest caz, jucătorul 2 este câștigătorul.

Se observă că au fost tastate si alte valori inafara de 'z', fapt semnalat printr-un mesaj corespunzător.




```
1 Sa ne jucam Snakes and Ladders!
2 Ai apasat tasta '1'. Te rog apasa 'z'!
3 Ai apasat tasta '2'. Te rog apasa 'z'!
4 Ai apasat tasta '3'. Te rog apasa 'z'!
5 Ai apasat tasta '4'. Te rog apasa 'z'!
6 Zarul jucatorului 1 indica valoarea 6.
7 Jucatorul 1 se afla pe pozitia 7.
8
9
10
11 Zarul jucatorului 2 indica valoarea 2.
12 Jucatorul 2 se afla pe pozitia 3.
13
14
15
16 Zarul jucatorului 1 indica valoarea 3.
17 Jucatorul 1 se afla pe pozitia 10.
18
19
20
21 Zarul jucatorului 2 indica valoarea 1.
22 Jucatorul 2 se afla pe pozitia 4.
23
24
25
26 Zarul jucatorului 134
27 Jucatorul 1 se 135
28 136
29 137
30 138
31 Zarul jucatorului 139
32 FELICITARI! Vei 140
33 Jucatorul 2 se 141
142
143
144
145 Zarul jucatorului 2 indica valoarea 4.
146 FELICITARI! Vei uita pe o scara :)
147 Jucatorul 2 se afla pe pozitia 100.
148
149
150
151 Jucatorul numarul 2 a castigat!!!
152
```



# A doua testare

În următoarele printscreen-uri realizate fișierului output2, este prezentată desfășurarea unei alte runde de joc, de data aceasta ieșind învingător jucătorul 1.

Observăm că acest joc a durat mai mult decât precedentul.

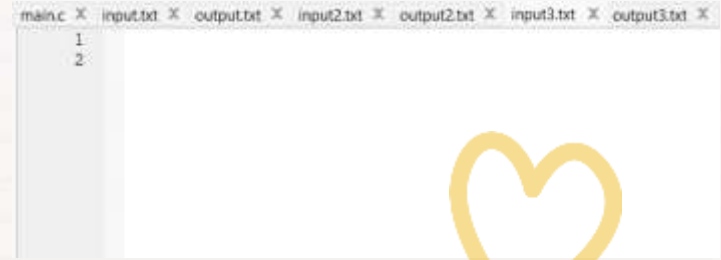


```
1 Sa ne jucam Snakes and Ladders!
2 zarul jucatorului 1 indica valoarea 4.
3 Jucatorul 1 se afla pe pozitia 5.
4
5
6
7 zarul jucatorului 2 indica valoarea 6.
8 Jucatorul 2 se afla pe pozitia 7.
9
10
11
12 zarul jucatorului 1 indica valoarea 4.
13 Jucatorul 1 se afla pe pozitia 9.
14
15
16
17 zarul jucatorului 2 indica valoarea 5.
18 Jucatorul 2 se afla pe pozitia 12.
19
20
21
22 zarul jucatorului 1 indica valoarea 2.
23 Jucatorul 1 se afla pe pozitia 11.
24
25
26
27 zarul jucatorului 2 indica valoarea 2.
28 Jucatorul 2 se afla pe pozitia 14.
29
30
31
32 zarul jucatorului 1 indica valoarea 6.
33 Jucatorul 1 se 243 zarul jucatorului 2 indica valoarea 5.
244 Jucatorul 2 se afla pe pozitia 45.
245
246
247
248 zarul jucatorului 1 indica valoarea 5.
249 Jucatorul 1 se afla pe pozitia 100.
250
251
252
253 Jucatorul numarul 1 a castigat!!!
254
```

# A treia testare

În cazul în care fișierul din care citim valorile 'z' necesare rulării programului este gol, ni se va afișa un mesaj corespunzător, conform conținutului fișierului output3.

Așadar, vom fi semnalați dacă oitem să introducem valorile corespunzătoare pentru testarea programului și vom ști exact că aceasta este eroarea.



```
1 Sa ne jucam Snakes and Ladders!  
2 Fișierul de intrare este gol.  
3
```



05

## Concluzii



- ✳ Simularea unui joc Snakes and Ladders poate fi realizată sub forma unui program C, reprezentând fiecare poziție pe care se poate afla un pion prin numărul său de ordine și folosind generarea random de numere pentru a obține efectul zarului.
- ✳ O abatere de la realitate este faptul că am folosit generarea de numere convenabile, pentru ca jocul să nu dureze mult prea mult.
- ✳ Testarea aplicației este necesară pentru a vedea felul în care aceasta gestionează seturi de date diferite și ajută la verificarea corectitudinii programului implementat.
- ✳ Realizarea schemelor logice pentru funcțiile importante a reprezentat un exercițiu interesant, care ne-a antrenat atenția asupra detaliilor și înțelegerii ordinii executării blocurilor de instrucțiuni.

# 06

## Appendix

```
if (sarpe_sau_scara != 0) //pozitia este una critica, vor avea loc
    modificari
{
    *pozitie += sarpe_sau_scara; //incrementam pozitia cu o valoare
    negativa sau pozitiva, in functie de context
    if (sarpe_sau_scara > 0) //verificam semnul numarului pentru a stii
    daca am avut de a face cu o SCARA sau un SARPE
        fprintf(out, "FELICITARI! Vei urca pe o scara :)\n");
    else
        fprintf(out, "UUUPS! Ai calcat pe un sarpe :(\n");
}
```

```
double calculeaza_bias(int pozitie)
{
    if (pozitie < 50)
        return 0.5; ///jucatorul are un punctaj mai mic de 50, deci se afla
        in prima jumatate a tablei => bias mai mic
    else
        return 0.5 + 0.4 * (pozitie - 50) / 50; ///biasul este crescut in
        functie de distanta de la mijlocul tablei
}
```

```
pozitia_jucatorului_2 = runda_noua(2, &pozitia_jucatorului_2, in,
    out); ///se modifica pozitia celui de-al doilea jucator
if (pozitia_jucatorului_2 == 100) //se verifica daca este egala cu
    100
{
    fprintf(out, "Jucatorul numarul 2 a castigat!!!\n"); //in caz
    afirmativ, se afiseaza un mesaj corespunzator
    break; //se iese din bucla
}
```





07

## References

- ✳️ <https://www.geeksforgeeks.org/basics-file-handling-c/>
- ✳️ <https://classplayground.com/printable-snakes-and-ladders-game/>
- ✳️ <https://stackoverflow.com/questions/52480128/biased-random-number-generator-function-in-c>
- ✳️ <https://stackoverflow.com/questions/21267716/error-handling-in-file-opening>





**Vă  
mulțumim!**