

# Proiect Seminar 2023-2024

## Programarea calculatoarelor si limbaje de programare

### SNAKES AND LADDERS

SPĂTĂRELU EDUARD-ȘTEFAN  
STICLAN DAIANA-VALENTINA

Ianuarie 2024

## 1 Introducere

### 1.1 Enuntul 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.

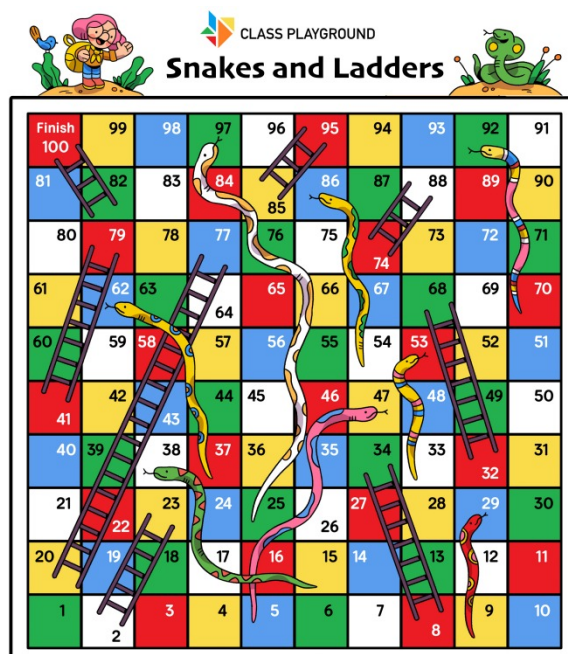
Șerpii îți scad punctele în timp ce scara le mărește.

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

Punctele reținute în variabile reprezintă de fapt poziții pe tabla de Snakes and Ladders. Fiecare jucător începe de pe poziția 1, prima poziție de pe tablă, avansând în funcție de valoarea indicată de zar. Aruncarea zarului se face succesiv de către cei doi participanți. Când poziția unui jucător corespunde unei scări, acesta va avansa până la capătul scării (valoarea poziției va crește). Când poziția unui jucător corespunde capătului unui șarpe, acesta se va întoarce pe tablă câte poziții îi indică șarpele. Pentru a vizualiza tabla de joc, am utilizat următorul șablon:



Conform tablei de joc aleasă:

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

## 2 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 files

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

#### END FUNCTION

#### 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

#### 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**

```

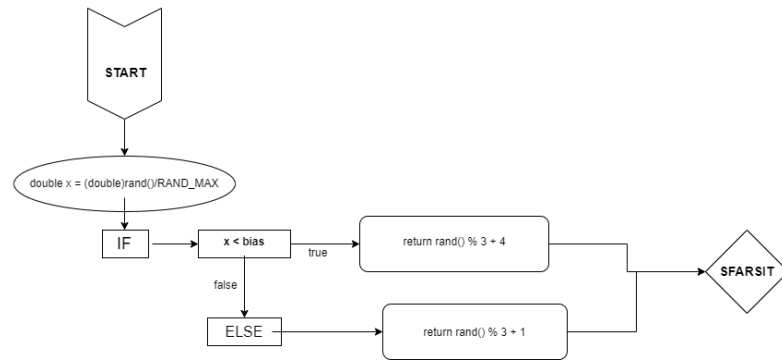
Read character from input file WHILE character is not 'z' DO
    IF character is EOF THEN
        Print error message and exit program
    ELSE IF character is not a newline THEN
        Print message asking for 'z'
    END IF
    Read next character from input file
END WHILE

```

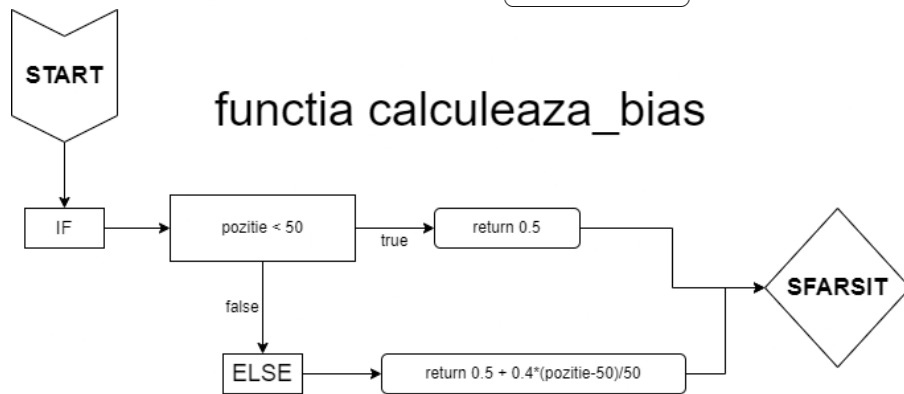
**END FUNCTION**

## 2.2 Scheme logice

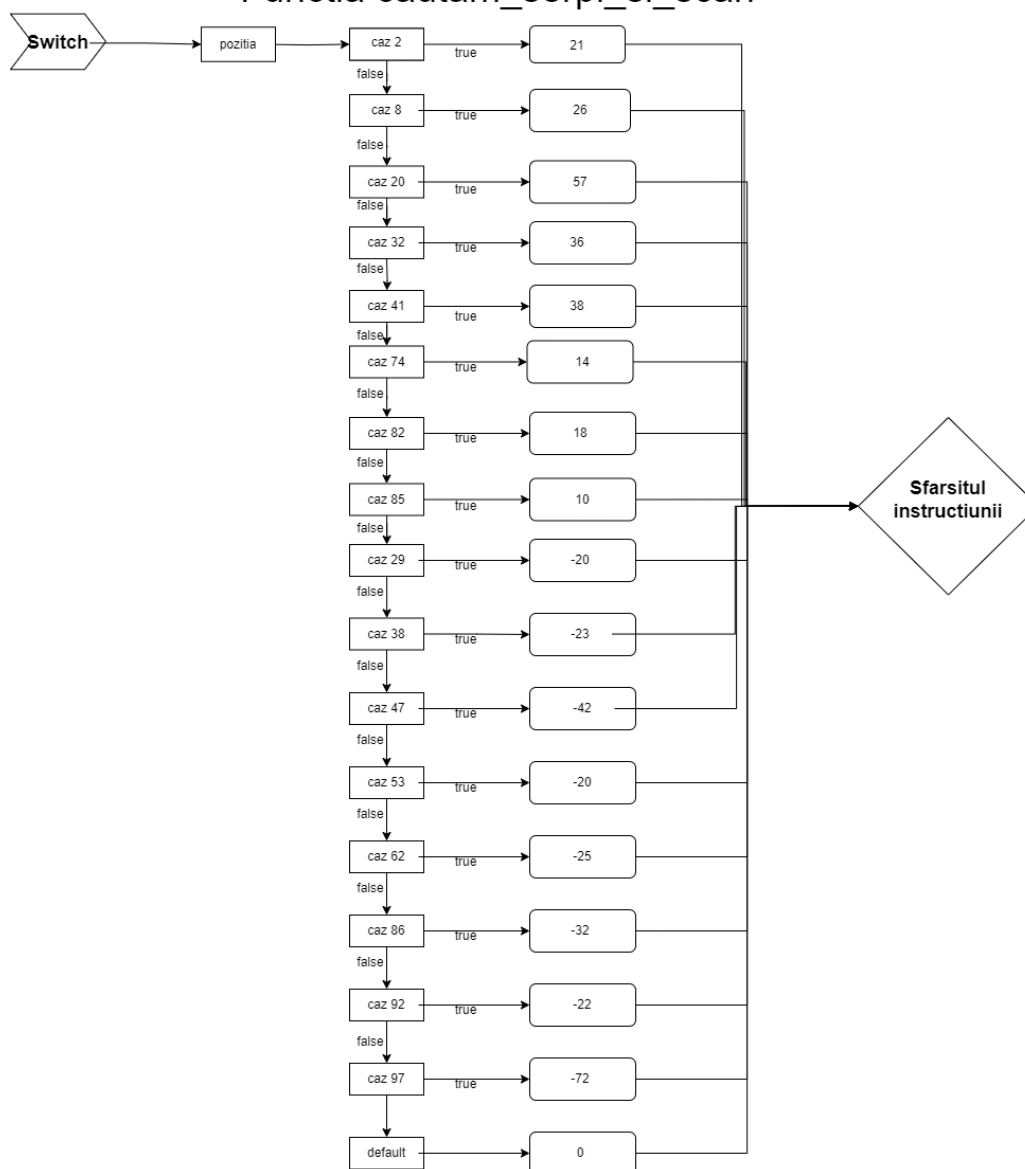
functia arunca\_zarul\_biasat



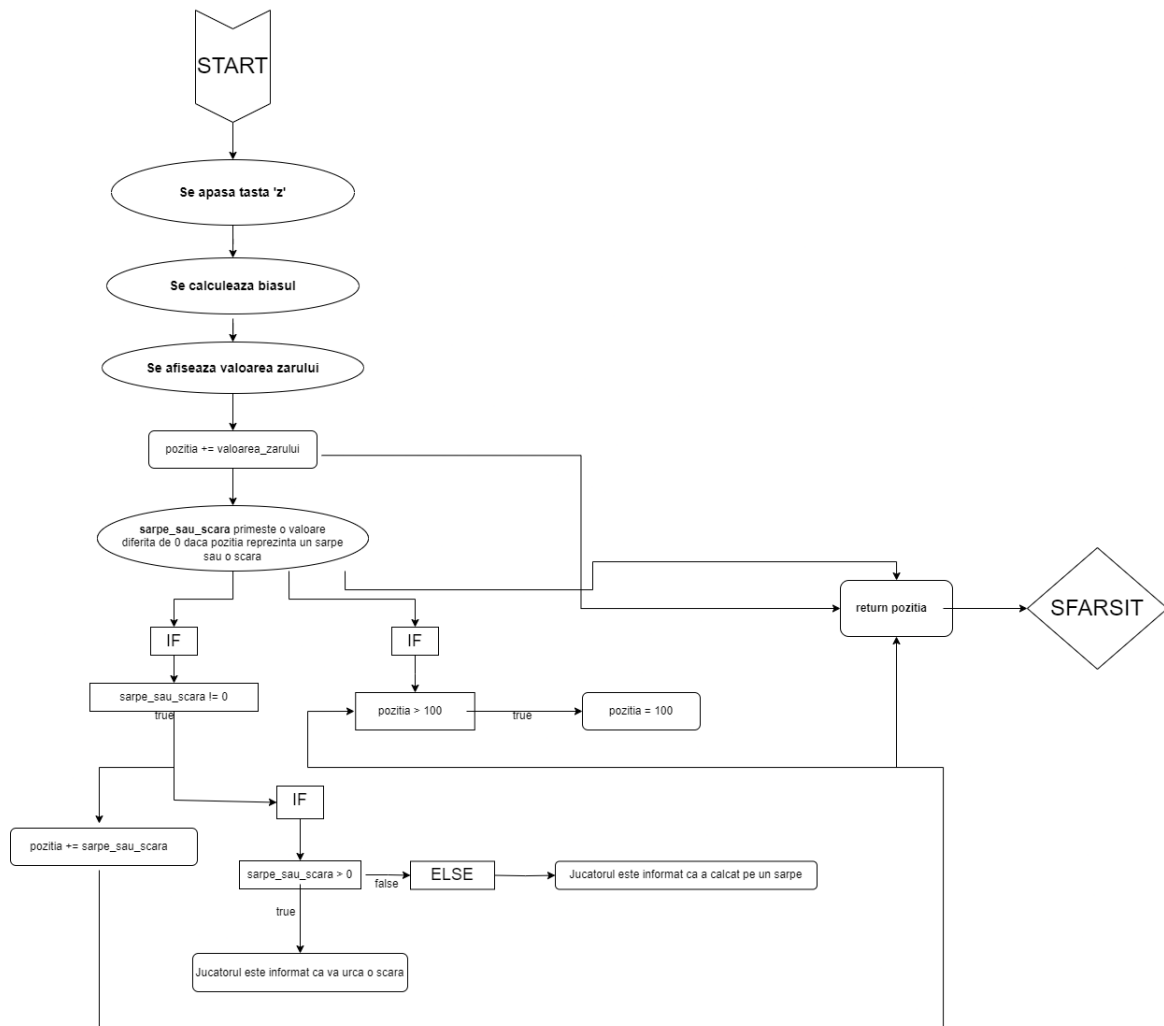
functia calculeaza\_bias



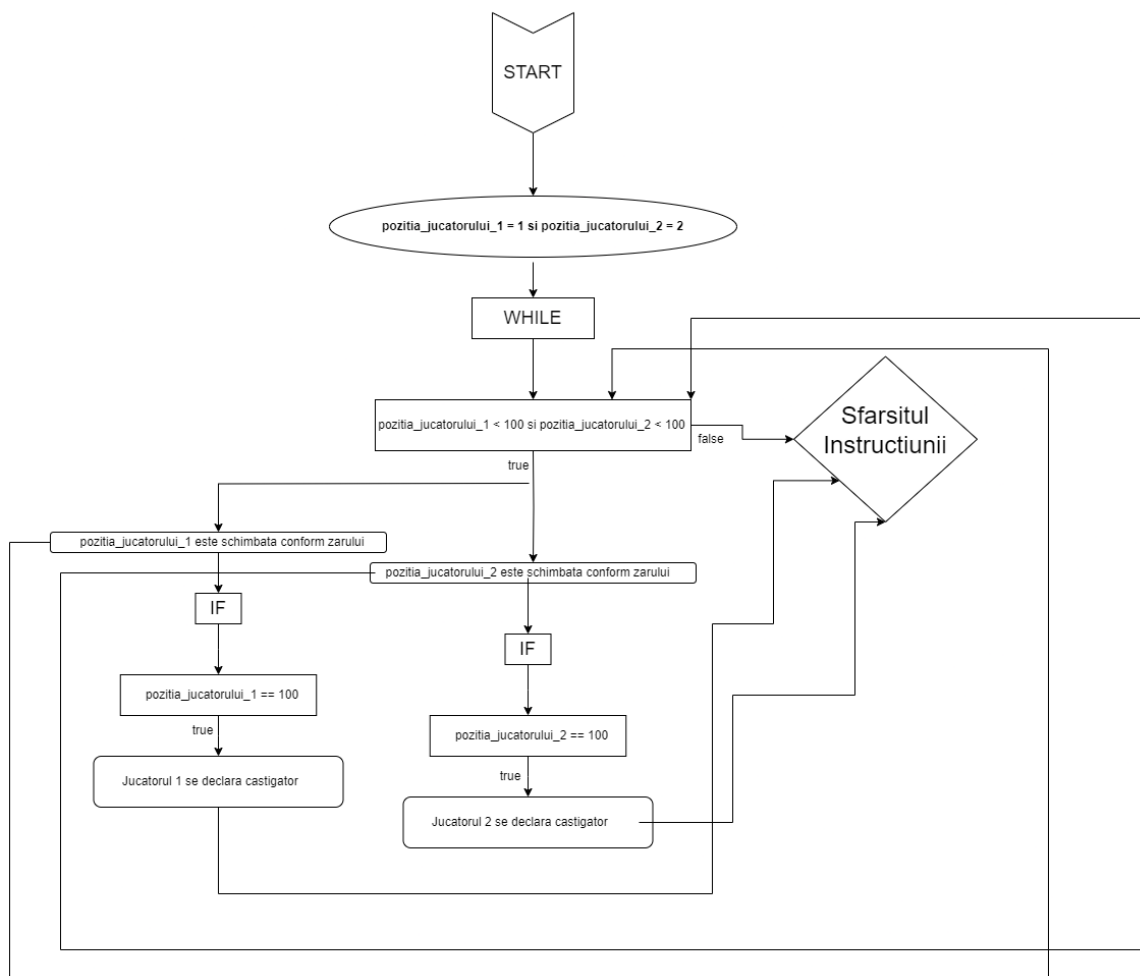
## Funcția cautam\_serpi\_si\_scari



## functia runda\_noua



## functia sa\_jucam\_jocul



## 3 Descrierea aplicatiei

### 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, exista o functie 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.

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 funcții 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.

## 4 Rezultate

Cum programul are la bază generarea de numere random, singura dată de intrare semnificativă pentru parcurgerea jocului este caracterul z.

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 și alte valori înafara 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 1 indica valoarea 1.
27 Jucatorul 1 se afla pe pozitia 11.
28
29
30
31 Zarul jucatorului 2 indica valoarea 4.
32 FELICITARI! Vei urca pe o scara :)
33 Jucatorul 2 se afla pe pozitia 34.
```



```
31 Zarul jucatorului 2 indica valoarea 4.
32 FELICITARI! Vei urca pe o scara :)
33 Jucatorul 2 se afla pe pozitia 34.
34
35
36
37 Zarul jucatorului 1 indica valoarea 6.
38 Jucatorul 1 se afla pe pozitia 17.
39
40
41
42 Zarul jucatorului 2 indica valoarea 6.
43 Jucatorul 2 se afla pe pozitia 40.
44
45
46
47 Zarul jucatorului 1 indica valoarea 5.
48 Jucatorul 1 se afla pe pozitia 22.
49
50
51
52 Zarul jucatorului 2 indica valoarea 5.
53 Jucatorul 2 se afla pe pozitia 45.
54
55
56
57 Ai apasat tasta '1'. Te rog apasa 'z'!
58 Ai apasat tasta '2'. Te rog apasa 'z'!
59 Ai apasat tasta '3'. Te rog apasa 'z'!
60 Zarul jucatorului 1 indica valoarea 1.
61 Jucatorul 1 se afla pe pozitia 23.
62
63
```

65	Zarul jucatorului 2 indica valoarea 5.
66	Jucatorul 2 se afla pe pozitia 50.
67	
68	
69	
70	Zarul jucatorului 1 indica valoarea 3.
71	Jucatorul 1 se afla pe pozitia 26.
72	
73	
74	
75	Zarul jucatorului 2 indica valoarea 1.
76	Jucatorul 2 se afla pe pozitia 51.
77	
78	
79	
80	Zarul jucatorului 1 indica valoarea 2.
81	Jucatorul 1 se afla pe pozitia 28.
82	
83	
84	
85	Zarul jucatorului 2 indica valoarea 5.
86	Jucatorul 2 se afla pe pozitia 56.
87	
88	
89	
90	Zarul jucatorului 1 indica valoarea 3.
91	Jucatorul 1 se afla pe pozitia 31.
92	
93	
94	
95	Zarul jucatorului 2 indica valoarea 5.
96	Jucatorul 2 se afla pe pozitia 61.
97	

```

100  Zarul jucatorului 1 indica valoarea 3.
101  Jucatorul 1 se afla pe pozitia 34.
102
103
104
105  Zarul jucatorului 2 indica valoarea 6.
106  Jucatorul 2 se afla pe pozitia 67.
107
108
109
110  Zarul jucatorului 1 indica valoarea 1.
111  Jucatorul 1 se afla pe pozitia 35.
112
113
114
115  Zarul jucatorului 2 indica valoarea 1.
116  Jucatorul 2 se afla pe pozitia 68.
117
118
119
120  Zarul jucatorului 1 indica valoarea 4.
121  Jucatorul 1 se afla pe pozitia 39.
122
123
124
125  Zarul jucatorului 2 indica valoarea 4.
126  Jucatorul 2 se afla pe pozitia 72.
127
128
129
130  Zarul jucatorului 1 indica valoarea 4.
131  Jucatorul 1 se afla pe pozitia 43.
132
133
134
135  Zarul jucatorului 2 indica valoarea 4.
136  Jucatorul 2 se afla pe pozitia 76.
137
138
139
140  Zarul jucatorului 1 indica valoarea 2.
141  Jucatorul 1 se afla pe pozitia 45.
142
143
144
145  Zarul jucatorului 2 indica valoarea 6.
146  FELICITARI! Vei urca pe o scara :)
147  Jucatorul 2 se afla pe pozitia 100.
148
149
150
151  Jucatorul numarul 2 a castigat!!!
152

```

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 afla pe pozitia 17.
```

31  
32 Zarul jucatorului 1 indica valoarea 6.  
33 Jucatorul 1 se afla pe pozitia 17.  
34  
35  
36  
37 Zarul jucatorului 2 indica valoarea 6.  
38 FELICITARI! Vei urca pe o scara :)  
39 Jucatorul 2 se afla pe pozitia 77.  
40  
41  
42  
43 Zarul jucatorului 1 indica valoarea 1.  
44 Jucatorul 1 se afla pe pozitia 18.  
45  
46  
47  
48 Zarul jucatorului 2 indica valoarea 6.  
49 Jucatorul 2 se afla pe pozitia 83.  
50  
51  
52  
53 Zarul jucatorului 1 indica valoarea 1.  
54 Jucatorul 1 se afla pe pozitia 19.  
55  
56  
57  
58 Zarul jucatorului 2 indica valoarea 4.  
59 Jucatorul 2 se afla pe pozitia 87.  
60

63 Zarul jucatorului 1 indica valoarea 2.  
64 Jucatorul 1 se afla pe pozitia 21.  
65  
66  
67  
68 Zarul jucatorului 2 indica valoarea 5.  
69 UUUPS! Ai calcat pe un sarpe :(  
70 Jucatorul 2 se afla pe pozitia 70.  
71  
72  
73  
74 Zarul jucatorului 1 indica valoarea 1.  
75 Jucatorul 1 se afla pe pozitia 22.  
76  
77  
78  
79 Zarul jucatorului 2 indica valoarea 5.  
80 Jucatorul 2 se afla pe pozitia 75.  
81  
82  
83  
84 Zarul jucatorului 1 indica valoarea 2.  
85 Jucatorul 1 se afla pe pozitia 24.  
86  
87  
88  
89 Zarul jucatorului 2 indica valoarea 4.  
90 Jucatorul 2 se afla pe pozitia 79.  
91  
92  
93  
94 Zarul jucatorului 1 indica valoarea 6.

93  
94 Zarul jucatorului 1 indica valoarea 6.  
95 Jucatorul 1 se afla pe pozitia 30.  
96  
97  
98  
99 Zarul jucatorului 2 indica valoarea 5.  
100 Jucatorul 2 se afla pe pozitia 84.  
101  
102  
103  
104 Zarul jucatorului 1 indica valoarea 4.  
105 Jucatorul 1 se afla pe pozitia 34.  
106  
107  
108  
109 Zarul jucatorului 2 indica valoarea 3.  
110 Jucatorul 2 se afla pe pozitia 87.  
111  
112  
113  
114 Zarul jucatorului 1 indica valoarea 5.  
115 Jucatorul 1 se afla pe pozitia 39.  
116  
117  
118  
119 Zarul jucatorului 2 indica valoarea 4.  
120 Jucatorul 2 se afla pe pozitia 91.  
121  
122  
123  
124 Zarul jucatorului 1 indica valoarea 5.

123  
 124 Zarul jucatorului 1 indica valoarea 5.  
 125 Jucatorul 1 se afla pe pozitia 44.  
 126  
 127  
 128  
 129 Zarul jucatorului 2 indica valoarea 6.  
 130 UUUPS! Ai calcat pe un sarpe :(  
 131 Jucatorul 2 se afla pe pozitia 25.  
 132  
 133  
 134  
 135 Zarul jucatorului 1 indica valoarea 4.  
 136 Jucatorul 1 se afla pe pozitia 48.  
 137  
 138  
 139  
 140 Zarul jucatorului 2 indica valoarea 5.  
 141 Jucatorul 2 se afla pe pozitia 30.  
 142  
 143  
 144  
 145 Zarul jucatorului 1 indica valoarea 6.  
 146 Jucatorul 1 se afla pe pozitia 54.  
 147  
 148  
 149  
 150 Zarul jucatorului 2 indica valoarea 6.  
 151 Jucatorul 2 se afla pe pozitia 36.  
 152  
 153  
 ...  
 155 Zarul jucatorului 1 indica valoarea 6.  
 156 Jucatorul 1 se afla pe pozitia 60.  
 157  
 158  
 159  
 160 Zarul jucatorului 2 indica valoarea 1.  
 161 Jucatorul 2 se afla pe pozitia 37.  
 162  
 163  
 164  
 165 Zarul jucatorului 1 indica valoarea 5.  
 166 Jucatorul 1 se afla pe pozitia 65.  
 167  
 168  
 169  
 170 Zarul jucatorului 2 indica valoarea 6.  
 171 Jucatorul 2 se afla pe pozitia 43.  
 172  
 173  
 174  
 175 Zarul jucatorului 1 indica valoarea 4.  
 176 Jucatorul 1 se afla pe pozitia 69.  
 177  
 178  
 179  
 180 Zarul jucatorului 2 indica valoarea 2.  
 181 Jucatorul 2 se afla pe pozitia 45.  
 182  
 183  
 184  
 185 Zarul jucatorului 1 indica valoarea 2.



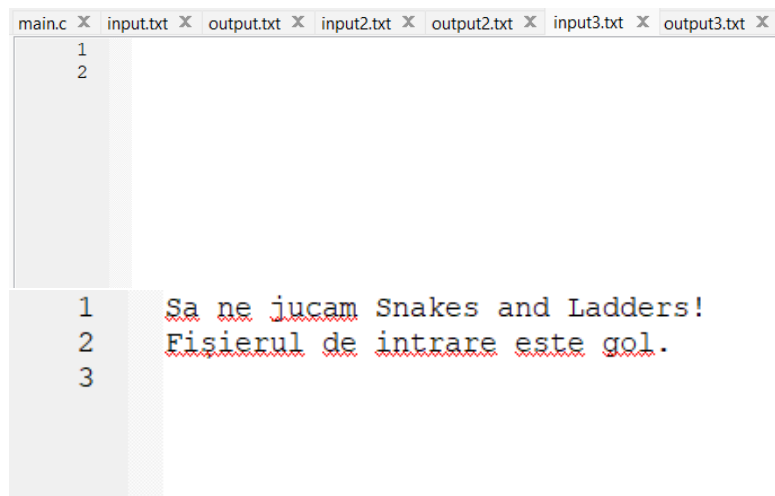
185 Zarul jucatorului 1 indica valoarea 2.  
 186 Jucatorul 1 se afla pe pozitia 71.  
 187  
 188  
 189  
 190 Zarul jucatorului 2 indica valoarea 1.  
 191 Jucatorul 2 se afla pe pozitia 46.  
 192  
 193  
 194  
 195 Zarul jucatorului 1 indica valoarea 3.  
 196 FELICITARI! Vei urca pe o scara :)  
 197 Jucatorul 1 se afla pe pozitia 88.  
 198  
 199  
 200  
 201 Zarul jucatorului 2 indica valoarea 4.  
 202 Jucatorul 2 se afla pe pozitia 50.  
 203  
 204  
 205  
 206 Zarul jucatorului 1 indica valoarea 4.  
 207 UUUPS! Ai calcat pe un sarpe :(  
 208 Jucatorul 1 se afla pe pozitia 70.  
 209  
 210  
 211  
 212 Zarul jucatorului 2 indica valoarea 6.  
 213 Jucatorul 2 se afla pe pozitia 56.  
 214  
 215  
 217 Zarul jucatorului 1 indica valoarea 4.  
 218 FELICITARI! Vei urca pe o scara :)  
 219 Jucatorul 1 se afla pe pozitia 88.  
 220  
 221  
 222  
 223 Zarul jucatorului 2 indica valoarea 3.  
 224 Jucatorul 2 se afla pe pozitia 59.  
 225  
 226  
 227  
 228 Zarul jucatorului 1 indica valoarea 6.  
 229 Jucatorul 1 se afla pe pozitia 94.  
 230  
 231  
 232  
 233 Zarul jucatorului 2 indica valoarea 1.  
 234 Jucatorul 2 se afla pe pozitia 60.  
 235  
 236  
 237  
 238 Zarul jucatorului 1 indica valoarea 4.  
 239 Jucatorul 1 se afla pe pozitia 98.  
 240  
 241  
 242  
 243 Zarul jucatorului 2 indica valoarea 5.  
 244 Jucatorul 2 se afla pe pozitia 65.  
 245

```

243 Zarul jucatorului 2 indica valoarea 5.
244 Jucatorul 2 se afla pe pozitia 65.
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

```

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 semnalăți dacă ometem să introducem valorile corespunzătoare pentru testarea programului și vom ști exact că aceasta este eroarea.



```

main.c x input.txt x output.txt x input2.txt x output2.txt x input3.txt x output3.txt x
1
2

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

```

## 5 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.

Jocul a fost supus unui proces de testare, așa cum se poate vedea în secțiunea de mai sus care conține rezultatele pentru 3 date de intrare diferite.

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.

## 6 Appendix: Program C

```

1 ///generarea unui numar aleator intre 1 si 6, dar cu o probabilitate mai
   mare de a genera
2 ///un numar mai mare daca jucatorul este mai aproape de sfarsitul jocului
3 int arunca_zarul_biasat(double bias)
4 {
5     //RAND_MAX este o constanta definita in biblioteca <stdlib.h> si
       reprezinta
6     //valoarea maxima care poate fi generata de functia rand()
7     double x = (double)rand() / RAND_MAX; //genereaza un numar aleator intre
       0 si 1, prin impartirea unui numar generat clasic, la valoarea
       maxima care poate fi generata
8     if (x < bias)

```

```

9         return rand() % 3 + 4; //daca x este mai mic decat bias, se
           genereaza un numar intre 4 si 6
10     else
11         return rand() % 3 + 1; //daca x este mai mare decat bias, se
           genereaza un numar intre 1 si 3
12 }

```

```

1  ///se calculeaza biasul pentru generarea numerelor random, in functie de
   pozitia jucatorului pe tabla
2  ///biasul creste odata cu apropierea de sfarsitul jocului, facilitand
   progresul jucatorilor
3  double calculeaza_bias(int pozitie)
4  {
5      if (pozitie < 50)
6          return 0.5; ///jucatorul are un punctaj mai mic de 50, deci se afla
           in prima jumatate a tablei => bias mai mic
7      else
8          return 0.5 + 0.4 * (pozitie - 50) / 50; ///biasul este crescut in
           functie de distanta de la mijlocul tablei
9  }

```

```

1  int cautam_serpi_si_scari(int pozitia)
2  {
3      switch(pozitia)
4      {
5          case 2:
6              return 21; //2->23, adica pe pozitia 2 se afla o scara, deci
                inaintam 21 de patratele pentru a ajunge pe pozitia 23
7          case 8:
8              return 26; //8->34
9          case 20:
10             return 57; //20->77
11          case 32:
12             return 36; //32->68
13          case 41:
14             return 38; //41->79
15          case 74:
16             return 14; //74->88
17          case 82:
18             return 18; //82->100
19          case 85:
20             return 10; //85->95
21          case 29:
22             return -20; //29->9, adica pe pozitia 29 se afla un sarpe si trebuie
                sa ne intoarcem 20 de patratele pentru a ajunge pe pozitia 9
23          case 38:
24             return -23; //38->15
25          case 47:
26             return -42; //47->5
27          case 53:
28             return -20; //53->33
29          case 62:
30             return -25; //62->37
31          case 86:
32             return -32; //86->54
33          case 92:
34             return -22; //92->70
35          case 97:
36             return -72; //97->25
37          default:

```

```

38     return 0;
39 }
40 }

```

```

1  int runda_noua(int nr_jucatorului, int *pozitia, FILE *in, FILE *out)
2  {
3      int valoarea_zarului, sarpe_sau_scara;
4
5      asteapta_apasare_z(in, out); //tasta z apasata de jucator pentru a
        inainta pe tabla
6
7      double bias = calculeaza_bias(*pozitia); //apelam functia necesara
        calcularii biasului
8      valoarea_zarului = arunca_zarul_biasat(bias); //generam numarul random
        care reprezinta de fapt valoarea afisata de zar
9      fprintf(out, "Zarul jucatorului %d indica valoarea %d.\n",
        nr_jucatorului, valoarea_zarului); //afisam valoarea zarului, in
        functie de jucator
10     *pozitia += valoarea_zarului; //incrementam pozitia
11
12
13     //urmeaza sa verificam valorile critice si sa incrementam sau
        decrementam pozitia dupa caz
14     sarpe_sau_scara = cautam_serpi_si_scari(*pozitia); //variabila primeste
        valoarea returnata de functie in cazul unei SCARI sau al unui SARPE
15
16     if (sarpe_sau_scara != 0) //pozitia este una critica, vor avea loc
        modificari
17     {
18         *pozitia += sarpe_sau_scara; //incrementam pozitia cu o valoare
            negativa sau pozitiva, in functie de context
19         if (sarpe_sau_scara > 0) //verificam semnul numarului pentru a stii
            daca am avut de a face cu o SCARA sau un SARPE
20             fprintf(out, "FELICITARI! Vei urca pe o scara :)\n");
21         else
22             fprintf(out, "UUUPS! Ai calcat pe un sarpe :(\n");
23     }
24
25     //daca scorul jucatorului devine mai mare decat 100, ne asiguram ca *
        pozitia va avea valoarea 100, pozitia finala de pe tabla din punct de
        vedere logic
26     if (*pozitia > 100)
27         *pozitia = 100;
28     fprintf(out, "Jucatorul %d se afla pe pozitia %d.\n", nr_jucatorului, *
        pozitia);
29     fprintf(out, "\n\n"); //aceste spatii determina o citire mai simpla a
        rezultatelor
30
31     return *pozitia;
32 }

```

```

1  void sa_jucam_jocul(FILE *in, FILE *out)
2  {
3      int pozitia_jucatorului_1 = 1, pozitia_jucatorului_2 = 1; //ambii
        jucatori pleaca de la aceeasi pozitie, adica cea de inceput
4
5      while (pozitia_jucatorului_1 < 100 && pozitia_jucatorului_2 < 100) //cat
        timp pozitiile sunt mai mici decat cea maxima, acestea vor fi
        modificate prin aruncarea zarului
6      {

```

```

7      pozitia_jucatorului_1 = runda_noua(1, &pozitia_jucatorului_1, in,
      out); ///se modifica pozitia primului jucator
8      if (pozitia_jucatorului_1 == 100) //se verifica daca este egala cu
      100
9      {
10         fprintf(out, "Jucatorul numarul 1 a castigat!!!\n"); //in caz
            afirmativ, se afiseaza un mesaj corespunzator
11         break; //se iese din bucla
12     }
13
14     pozitia_jucatorului_2 = runda_noua(2, &pozitia_jucatorului_2, in,
      out); ///se modifica pozitia celui de-al doilea jucator
15     if (pozitia_jucatorului_2 == 100) //se verifica daca este egala cu
      100
16     {
17         fprintf(out, "Jucatorul numarul 2 a castigat!!!\n"); //in caz
            afirmativ, se afiseaza un mesaj corespunzator
18         break; //se iese din bucla
19     }
20 }
21 }

```

## References

- [1] [GeeksforGeeks - Basics of File Handling in C](#)
- [2] [Class Playground - Printable Snakes and Ladders Game](#)
- [3] [Biased Random Number Generator Function in C](#)
- [4] [Using files](#)