

NETROM
premium
software



NETROM Code Golf

EDIȚIA a II-a



Cât e scorul?

Scrie o funcție care calculează scorul unui set de tenis de câmp (un singur set) ce se desfășoară după principiul „Advantage set” (în care e declarat câștigător jucătorul ce ajunge la minim 6 game-uri și o diferență de minim 2 game-uri față de adversar). Funcția primește ca parametru indicele jucătorului care a punctat (1 sau 2) și afișează în consolă scorul curent și jucătorul care a câștigat, în cazul în care setul s-a terminat. Funcția se poate folosi de o singură variabilă globală „a”, de tip **unsigned long int**, inițializată cu 0.

Ieșirea (output-ul) trebuie să respecte următorul format:

Games: <Games player 1>-<Games player 2>

Points: <Points player 1>-<Points player 2>

În cadrul unui game scorul se calculează astfel:

- nici un punct: 0
- primul punct: 15
- al doilea punct: 30
- al treilea punct: 40
- la al cincilea punct se adjudecă game-ul și începe unul nou dacă adversarul nu a făcut 40, altfel se joacă la avantaj
- jocul la avantaj presupune câștigarea game-ului de către jucătorul care are o diferență de 2 puncte. Atunci când se joacă la avantaj, se va afișa „Ad” pentru jucătorul care este în avantaj și 40 pentru celălalt jucător. În cazul în care nici un jucător nu este în avantaj se va afișa 40 la ambii jucători.

Din momentul în care setul s-a încheiat, funcția o să afișeze în plus și indicele jucătorului câștigător, în formatul: „Winner 1” sau „Winner 2”.

Exemple de output:

1. Games: 0-0
Points: 15-0
2. Games: 1-0
Points: 15-30
3. Games: 8-7
Points: Ad-40
4. Games: 6-4
Points: 0-0
Winner 1

Observații: Testarea se va face prin apelarea succesivă a funcției cu diverse valori ale parametrului de intrare, de maximum 100 de ori în cadrul unui ciclu de testare.

Prototip funcție:

```
void f(unsigned char w)
```

unde:

- w reprezintă jucătorul care a punctat: w=1 a punctat jucătorul 1; w=2 a punctat jucătorul 2.

Exemplu:

Input: w=1, Output:

Games: 0-0
Points: 15-0

Input: w=1, Output:

Games: 0-0
Points: 30-0

Input: w=1, Output:

Games: 0-0
Points: 40-0

Input: w=1, Output:

Games: 1-0
Points: 0-0

Input: w=2, Output:

Games: 1-0
Points: 0-15

Cutia de viteze

Scie o funcție care validează mișcările brațului mecanic al unei cutii de viteze manuală. Acesta se poate muta dintr-o poziție dată într-o poziție viitoare printr-o mișcare de tip: înainte, înapoi, stânga sau dreapta. Funcția primește la intrare descrierea cutiei de viteze și secvența de mișcări plecând din poziția N (neutru).

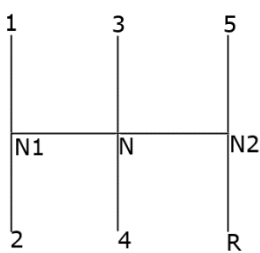
Descrierea cutiei de viteze se face prin specificarea pozițiilor posibile ale schimbătorului, iar pentru fiecare poziție în parte, se descriu posibilele poziții următoare, specificându-se codul mișcării care determină respectiva schimbare a poziției. Descrierea este un șir de șiruri de caractere, în care:

- primul șir conține pozițiile posibile (maxim 2 caractere pe poziție), separate prin spațiu (un nume de poziție nu poate conține spații);
- următoarele șiruri reprezintă mișcările posibile (o mișcare în fiecare șir), în formatul <Poziție 1> <simbol mișcare> <Poziție 2>, unde simbolul de mișcare poate fi unul din caracterele: U(p), D(own), L(eft), R(ight).

Secvența de mișcări ale brațului mecanic, care trebuie validată de funcție, se specifică printr-un șir de caractere, în care fiecare caracter reprezintă o mișcare, în formatul <simbol1><simbol2><simbolX>.

Funcția trebuie să tipărească în consolă „Valid, Poziția: ” și poziția finală a brațului mecanic, dacă secvența de mișcări este validă, sau mesajul „Invalid” în caz contrar.

În exemplul de mai jos este descrisă o cutie de viteze cu 9 poziții (1,3, 5, N1, N, N2, 2, 4 și R) care suportă următoarele mișcări:



N cu mișcarea L(Left) trece în poziția N1
 N cu mișcarea R(Rigth) trece în poziția N2
 N cu mișcarea U(Up) trece în poziția 3
 N cu mișcarea D(Down) trece în poziția 4
 N1 cu mișcarea U(Up) trece în poziția 1
 N1 cu mișcarea D(Down) trece în poziția 2
 N1 cu mișcarea R(Right) trece în poziția N
 N2 cu mișcarea U(Up) trece în poziția 5
 N2 cu mișcarea D(Down) trece în poziția R
 N2 cu mișcarea L(Left) trece în poziția N
 1 cu mișcarea D(Down) trece în poziția N1
 2 cu mișcarea U(Up) trece în poziția N1
 3 cu mișcarea D(Down) trece în poziția N
 4 cu mișcarea U(Up) trece în poziția N
 5 cu mișcarea D(Down) trece în poziția N2
 R cu mișcarea U(Up) trece în poziția N2

Observație: se consideră că parametrii funcției conțin o descriere validă a cutiei de viteze:

- mișcările sunt doar una din cele 4 descrise mai sus (D, U, L, R);
- pentru o poziție curentă și o anumită mișcare este specificată o singură poziție următoare;
- nu vor fi folosite poziții inexistente în descrierea mișcărilor

Prototip funcție:

```
void f(char* cv[], char l, char* t)
```

unde:

- *cv* reprezintă descrierea cutiei de viteze (în formatul specificat);
- *l* reprezintă numărul de elemente din șirul de șiruri de caractere (null terminated);
- *t* reprezintă șirul de tranziții de verificat.

Exemplu:

CV=

```

V1 V2 V3 R N
N U V1
N D R
N R V3
N L V2
V2 R N
V1 D N
V3 L N
R U N

```

L=9

Mișcări: UDLRR

Output: Valid, Poziția: V3

Mișcări: UDDL

Output: Invalid

Mișcări: DD

Output: Invalid

Afișare compactă

Scrive o funcție care primește la intrare un șir de numere întregi pe care-l tipărește în consolă în ordine crescătoare, separând prin virgulă elementele lui și înlocuind orice succesiune de mai mult de 2 numere consecutive cu notația intervalelor de numere (de ex. succesiunea de numere 4,5,6 se rescrie în formatul 4-6).

Prototip funcție:

```
void f(short int s[], unsigned short int l)
```

unde:

s[] reprezintă șirul de numere întregi

l reprezintă lungimea șirului *s*

Exemplu:

```
s = {-1,-2,-3,5,6,21,20,22,23,24,43}
```

```
l=11
```

```
Output:-3--1,5,6,20-24,43
```