

Il gioco di Ioiwari

Il problema

La famiglia di giochi, cosiddetti Mancala, che si giocano usando palline e buche è una delle più antiche forme di intrattenimento umano. In questo problema parliamo di una speciale versione di questi giochi inventata specificamente per le IOI.

Il gioco si svolge fra due giocatori su una tavola rotonda con sette buche disposte sul bordo; oltre alle buche, ogni giocatore possiede una banca. All'inizio del gioco vengono distribuite a caso 20 palline nelle buche in modo tale che ogni buca contenga almeno 2 e al massimo 4 palline. I due giocatori muovono a turno. In ciascuna mossa, il giocatore che deve muovere sceglie una buca non vuota ed estrae dalla buca tutte le palline che contiene, mettendosele in mano. A questo punto, finché la mano del giocatore contiene qualche pallina, il giocatore considera le buche una dopo l'altra, in senso orario, a partire da quella che segue immediatamente la buca che è stata svuotata, effettuando le seguenti operazioni:

- Se nella mano c'è più di una pallina: se la buca corrente contiene già 5 palline, il giocatore estrae una pallina dalla buca corrente e la mette nella propria banca; altrimenti, toglie una pallina dalla propria mano e la mette nella buca corrente;
- Se nella mano c'è una pallina: se la buca corrente contiene almeno 1 e al massimo 4 palline, il giocatore prende tutte le palline che ci sono nella buca e quella che ha in mano e le mette nella propria banca; altrimenti (se la buca contiene 0 o 5 palline) mette la pallina che ha in mano nella banca dell'avversario.

Il gioco termina quando tutte le buche sono vuote, e il vincitore è il giocatore che ha più palline nella sua banca.

Il giocatore che muove per primo ha una strategia vincente. Dovete scrivere un programma che giochi a Ioiwari come giocatore iniziale e vinca sempre. L'avversario giocherà in modo ottimale, cioè, ogni volta che gliene venga data la possibilità, riuscirà a vincere e il vostro programma verrà sconfitto.

Dati in input e output

Il vostro programma deve leggere dallo standard input e scrivere sullo standard output. Il vostro programma è il giocatore 1, e l'avversario è il giocatore 2. Quando il programma inizia, deve in primo luogo leggere una riga contenente 7 interi,

p_1, \dots, p_7 che rappresentano il numero di palline inizialmente presenti nelle buche $1, \dots, 7$, rispettivamente: le buche sono numerate con interi da 1 a 7 in senso orario.

Dopo aver fatto questo, il gioco ha inizio con le banche vuote. Il vostro programma deve giocare come segue:

- quando è il suo turno, il programma deve scrivere sullo standard output il numero della buca che definisce la sua mossa (cioè quella che viene svuotata);
- quando è il turno dell'avversario, deve leggere dallo standard input il numero della buca che definisce la mossa dell'avversario.

Strumenti

Vi viene dato un programma (`ioiwari2` sotto Linux, `ioiwari2.exe` sotto Windows) che gioca da una certa posizione iniziale in modo ottimale come giocatore 2. Questo programma scrive per prima cosa sul suo standard output la prima riga che il vostro programma dovrebbe leggere, contenente il numero di palline nelle sette buche in quella posizione iniziale:

```
4 3 2 4 2 3 2
```

Dopo aver fatto questo, il programma giocherà come giocatore 2, leggendo le mosse del giocatore 1 dallo standard input e scrivendo le sue mosse sullo standard output. Potete provare il vostro programma e `ioiwari2` in due finestre separate copiando manualmente la conversazione tra i due programmi. La conversazione viene registrata nel file `ioiwari.out`.

Istruzioni di programmazione

Negli esempi sotto indicati, supponiamo che dobbiate leggere un intero nella variabile `last`, e che la variabile intera `mymove` contenga la vostra mossa.

Se programmate in C++ e usate `iostream`, dovete implementare la lettura e la scrittura come segue:

```
cout << mymove << endl << flush;
cin >> last;
```

Se programmate in C o C++ e usate `scanf` e `printf`, dovete implementare la lettura e la scrittura come segue:

```
printf("%d\n", mymove); fflush(stdout);
scanf("%d", &last);
```

Se programmate in Pascal, dovete implementare la lettura e la scrittura come segue:

```
Writeln(mymove);
Readln(last);
```

Esempio

Ecco una sequenza corretta di 6 mosse:

Mossa / Numero della buca	Contenuto delle buche							Banca 1	Banca 2
	1	2	3	4	5	6	7		
Situazione iniziale	4	3	2	4	2	3	2	0	0
Mossa del giocatore 1: 2	4	0	3	5	0	3	2	3	0
Mossa del giocatore 2: 3	4	0	0	4	1	4	0	3	4
Mossa del giocatore 1: 5	4	0	0	4	0	0	0	8	4
Mossa del giocatore 2: 4	0	0	0	0	1	1	1	8	9
Mossa del giocatore 1: 5	0	0	0	0	0	0	1	10	9
Mossa del giocatore 2: 7	0	0	0	0	0	0	0	11	9

Punteggio

Se il vostro programma vince in uno dei test, ottenete 4 punti per quel test; in caso di pareggio, ottenete 2 punti; altrimenti, ottenete 0 punti.