FIBONACCI

Denumirea subdirectorului: **FIBO**
Denumirea fișierului de intrare: **FIBO.IN**
Denumirea fișierului de ieșire: **FIBO.OUT**

Descrierea problemei

Se consideră celebrul șir al lui *Fibonacci*, definit astfel:
- $F_0 = 0$;
- $F_1 = 1$;
- $F_i = F_{i-1} + F_{i-2}$ pentru $i > 1$.

Sarcina voastră este de a determina indicele unui element al acestui șir care este divizibil cu un număr de forma 2^n.

Valoarea indicelului trebuie să fie mai mică decât 2^n. De exemplu, pentru $n = 3$, unul dintre elementele divizibile cu 2^3 este 8 (are indicele 6 și $6 < 8$).

Date de intrare

Fișierul de intrare conține o singură linie pe care se află valoarea n.

Date de ieșire

Fișierul de ieșire va conține o singură linie pe care se va afla indicele elementului divizibil cu 2^n.

În cazul în care nu există nici un element divizibil cu 2^n a cărui indice să aibă o valoare mai mică decât 2^n, atunci în fișierul de ieșire se va scrie valoarea -1.

Restricție

- $1 \le n \le 10000$.

Exemple

FIBO.IN	**FIBO.OUT**
3	6

FIBO.IN	**FIBO.OUT**
1	-1

FIBO.IN	**FIBO.OUT**
5	24

FIBO.IN	**FIBO.OUT**
2	-1

Explicație

Primii termeni ai șirului lui *Fibonacci* sunt:
- $F_0 = 0$;
- $F_1 = 1$;
- $F_2 = 0 + 1 = 1$;
- $F_3 = 1 + 1 = 2$;
- $F_4 = 1 + 2 = 3$;
- $F_5 = 2 + 3 = 5$;
- $F_6 = 3 + 5 = 8$;
- $F_7 = 5 + 8 = 13$;
- $F_8 = 8 + 13 = 21$;
- $F_9 = 13 + 21 = 34$;
- $F_{10} = 21 + 34 = 55$;
- $F_{11} = 34 + 55 = 89$;
- $F_{12} = 55 + 89 = 144$;
- $F_{13} = 89 + 144 = 233$;
- $F_{14} = 144 + 233 = 377$;
- $F_{15} = 233 + 377 = 610$;
- $F_{16} = 377 + 610 = 987$;
- $F_{17} = 610 + 987 = 1597$;
- $F_{18} = 987 + 1597 = 2584$;
- $F_{19} = 1597 + 2584 = 4181$;
- $F_{20} = 2584 + 4181 = 6765$;
- $F_{21} = 4181 + 6765 = 10946$;
- $F_{22} = 6765 + 10946 = 17711$;
- $F_{23} = 10946 + 17711 = 28657$;
- $F_{24} = 17711 + 28657 = 46368$.

Un element divizibil cu $2^3 = 8$ este 8; indicele acestuia are valoarea 6; această valoare este mai mică decât 8. Cel mai mic element divizibil cu $2^1 = 2$ este 2; indicele acestuia are valoarea 3; această valoare este mai mare decât 2. Un element divizibil cu $2^5 = 32$ este 46368; indicele acestuia are valoarea 24; această valoare este mai mică decât 32. Cel mai mic element divizibil cu $2^2 = 4$ este 8. Indicele acestuia are valoarea 6; această valoare este mai mare decât 4.

Timp de execuție/test: **2 secunde**

Runda 5, problema 2

ATLANTIS

Denumirea subdirectorului: **ATLANTIS**
Denumirea fișierului de intrare: **ATLANTIS.IN**
Denumirea fișierului de ieșire: **ATLANTIS.OUT**

Descrierea problemei

După ce a reușit să treacă de poarta cetății *Atlantis*, ***Iahim Uratrox***, marele explorator al *Atlantidei*, a descoperit o mulțime dintre vestigiile vechii civilizații.

Singura clădire în care nu a putut intra este palatul foștilor suverani atlanți. Se presupune că în interiorul acestui palat se pot găsi documentele care să explice motivul scufundării miticului oraș. Din nefericire, intrarea în palat este protejată și ea de un cifru.

Mecanismul este unul mult mai simplu, el constând din doar două dispozitive de introducere a codului. Pe mecanism este gravat, la fel ca și la intrare, un număr format din șase cifre.

Bineînțeles, savanții au reușit imediat să descopere algoritmul de funcționare al mecanismului. Cele două dispozitive corespund semnelor `'+'` și `'-'`. Fiecare atingere a unui dispozitiv duce la adunarea sau scăderea unui număr. Acest număr variază în funcție de numărul de ordine al atingerii. La a i-a atingere, numărul care se adună sau se scade este i^2. Se pornește cu valoarea 0.

Așadar, la fiecare atingere i valoarea corespunzătoare mecanismului scade sau crește cu i^2. În final, pentru a deschide poarta, trebuie să se obțină valoarea gravată pe mecanism.

Date de intrare

Fișierul de intrare conține o singură linie pe care se află cele șase cifre ale numărului gravat pe mecanismul cifrului, neseparate prin spații. Există posibilitatea ca prima, primele două, primele trei, primele patru sau primele cinci cifre să fie 0. Totuși, cel puțin una dintre cele șase cifre este diferită de zero.

Date de ieșire

Fișierul de ieșire va conține o singură linie pe care se vor semnele corespunzătoare atingerilor mecanismului. Al i-lea semn de pe această linie va corespunde celei de-a i-a atingeri. Dacă prin această atingere se adună valoarea i^2, atunci semnul va fi `'+'`. Dacă atingerea duce la scăderea valorii i^2, atunci semnul va fi `'-'`.

Restricții și precizări

- Valoarea numărului inscripționat pe mecanismul cifrului este cuprinsă între 1 și 999999.
- În cazul în care nu poate fi descoperită o secvență de atingeri, *Iahim Uratrox* nu va mai putea deschide poarta folosind o încărcătură explozivă așa cum ar fi putut face atunci când se afla în fața porții orașului, deoarece ar risca să distrugă prea multe vestigii importante; astfel, cel mai mare secret al *Atlantidei* va fi păstrat pentru totdeauna. Din fericire, după câteva luni, exploratorul s-a întors și a prezentat motivele scufundării *Continentului Pierdut*; se pare că explicația era atât de îngrijorătoare încât s-a decis ca motivele să nu fie cunoscute publicului. Totuși, concluzia evidentă care poate fi trasă este că există întotdeauna o secvență corectă de atingeri care duc la deschiderea porții.
- Poarta se deschide chiar dacă numărul atingerilor nu este minim.
- Dacă există mai multe soluții, doar una trebuie scrisă în fișierul de ieșire.

Exemple

ATLANTIS.IN	**ATLANTIS.OUT**
000012	++-+

ATLANTIS.IN	**ATLANTIS.OUT**
000055	+++++

Explicații

- $0 + 1^2 + 2^2 - 3^2 + 4^2 = 0 + 1 + 4 - 9 + 16 = 12$;
- $0 + 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 0 + 1 + 4 + 9 + 16 + 25 = 55$.

Timp de execuție/test: **5 secunde**

TIC-TAC-TOE

Denumirea subdirectorului: **TICTAC**

Denumirea fișierului de intrare: **-**

Denumirea fișierului de ieșire: **-**

Descrierea problemei

Se consideră o tablă de dimensiuni 3×3 care este inițial goală. Pe rând, jucătorii mută, marcând una din căsuțele libere cu un semn propriu. Prin convenție semnul primului jucător este `"X"`, iar semnul celui de-al doilea este `"0"`. Jucătorul care obține primul trei semne proprii pe aceeași linie, coloană sau diagonală este declarat câștigător. Dacă toate căsuțele au fost marcate și nici un jucător nu a câștigat atunci se declară ***remiză***.

Presupunând că primul jucător este reprezentat de calculator, se cere să se programeze o strategie pentru al doilea jucător. Calculatorul va realiza întotdeauna operații deterministe pentru a efectua o mutare.

Pentru fiecare partidă câștigată veți primi 0.002 puncte. Pentru fiecare remiză veți primi 0.001 puncte. Pentru o partidă pierdută nu veți primi nici un punct.

Va trebui ca, într-un interval de o secundă, să jucați cât mai multe partide. Timpul consumat de calculator pentru a determina pozițiile vă permite (în cazul în care veți folosi un algoritm eficient) să jucați cel puțin 100000 de partide.

Cele nouă căsuțe ale tablei vor fi numerotate astfel:

0	1	2
3	4	5
6	7	8

Programatorii în *Pascal* vor avea la dispoziție un *unit* numit **`TICTAC.PAS`**, care va conține două proceduri și o funcție.

La începutul jocului va trebui, obligatoriu, să fie apelată procedura `InitGame`, care nu are nici un parametru și este folosită pentru inițializarea jocului.

Pentru a efectua o mutare veți apela procedura `MyMove`. Aceasta are un singur parametru care indică poziția în care veți amplasa un `"0"`.

Pentru a afla mutarea efectuată de adversar veți apela funcția `YourMove`. Aceasta nu are nici un parametru și returnează poziția în care calculatorul a amplasat un `"X"`. În cazul în care jocul se încheie, funcția va returna valoarea `-1`.

Programatorii în *C/C++* vor avea la dispoziție un *header* numit **`TICTAC.H`**, care va conține trei funcții.

La începutul jocului va trebui, obligatoriu, să fie apelată funcția `InitGame`, care nu are nici un parametru și este folosită pentru inițializarea jocului.

Pentru a efectua o mutare veți apela funcția `MyMove`. Aceasta are un singur parametru care indică poziția în care veți amplasa un `"0"`.

Pentru a afla mutarea efectuată de adversar veți apela funcția `YourMove`. Aceasta nu are nici un parametru și returnează poziția în care calculatorul a amplasat un `"X"`. În cazul în care jocul se încheie, funcția va returna valoarea `-1`.

Desfășurarea unui joc este următoarea: la început se va inițializa jocul, după care se va cere efectuarea unei mutări de către adversar. Apoi, alternativ, se vor efectua mutări și se vor cere mutările adversarului până în momentul în care funcția `YourMove` va returna valoarea `-1`. Chiar dacă mutarea voastră este câștigătoare, va trebui să mai apelați o dată funcția `YourMove` care va returna, obligatoriu, valoarea `-1`. Următorul joc va începe după o nouă inițializare.

Funcția `YourMove` va întrerupe execuția programului în momentul în care va expira timpul alocat partidelor.

În cazul în care nu veți respecta secvența de mutări, veți pierde jocul respectiv. Această sitiuație poate apărea dacă:

- se efectuează două mutări consecutive;
- se cere efectuarea a două mutări consecutive;
- se efectuează sau se cer mutări fără a se inițializa jocul;
- se inițializează jocul înaintea terminării unui joc (înainte ca funcția `YourMove` să returneze valoarea `-1`);
- nu se cere efectuarea primei mutări de către advesar;
- orice altă situație în care nu se respectă regulile jocului.

Câte o variantă a *unit*-ului și a *header*-ului sunt disponibile pentru *download* la adresa **www.ginfo.ro/concurs/tic-tactoe.shtml**. Calculatorul va folosi o strategie aleatoare, deci acestea nu vor fi folosite în timpul evaluării.

Programul vostru nu va trebui să citească date din nici un fișier și nu va trebui să scrie date în nici un fișier.