Factorial

Câte cifre are $N!$ ($1 \leq N \leq 1.000.000$)?

Pe prima linie a fișierului `factorial.in` se găsește N, iar rezultatul se afișează în fișierul `factorial.out`.

Timp de execuție/test: 1 sec.

factorial.in	factorial.out	factorial.in	factorial.out	factorial.in	factorial.out
5	3	1234	3281	1000000	5565709

Furtuna magnetică

Undeva în spațiu se găsește o navetă spațială de mici dimensiuni de tip Puddle Jumper (PJ) creată de strămoși. Aceasta se poate deplasa într-un plan care se consideră caroiat având două dimensiuni. Sistemul de coordonate este cel cartezian. Naveta PJ pornește dintr-un punct de start $(x_{initial}, y_{initial})$ și trebuie să ajungă într-un punct (x_{final}, y_{final}), ambele date. În decursul unei zile naveta PJ se poate deplasa cu ajutorul forței propulsoarelor proprii de la coordonata curentă la una învecinată, pe una din patru direcții: `LEFT` (`L`), `RIGHT` (`R`), `UP` (`U`), `DOWN` (`D`). Totodată în decursul unei zile naveta PJ poate să nu-și pornească propulsoarele dând comanda `STOP` (`S`), caz în care ea rămâne la coordonata curentă.

Din cauza unor turbulențe, în sistemul solar se produce o furtună magnetică care durează mai multe zile, furtună care poate devia naveta PJ. În fiecare zi furtuna poate împinge naveta PJ în una din cele 4 direcții `LEFT`, `RIGHT`, `UP`, `DOWN`, cu o singură celulă pe zi. Vestea bună este că strămoșii au conceput un sistem de predicție 100% sigur a direcțiilor furtunii pentru fiecare zi, sistem care e disponibil la bordul navetei PJ. Totodată se cunoaște că furtuna se produce un număr finit de zile după care se instalează un calm magnetic în tot universul.

Mișcările navetei PJ se pot combina cu mișcările furtunii astfel:

i) dacă într-o zi se ia decizia de a se propulsa naveta PJ în direcția furtunii pe o coordonată, naveta se va deplasa câte 2 celule pe acea coordonată în acea zi;

ii) dacă se ia decizia de a se deplasa naveta PJ pe o coordonată și furtuna acționează pe cealaltă coordonată atunci naveta PJ se va deplasa oblic (pe diagonală), deci mișcările se pot compune;

iii) dacă se ia decizia de a se deplasa naveta PJ contrar direcției furtunii pe o coordonată atunci naveta PJ va sta pe loc pe acea coordonată;

iv) dacă naveta PJ își închide propulsoarele, ea se va deplasa cu o celulă în direcția dată de furtună în acea zi, dacă evident furtuna nu a luat sfârșit.

Datele de intrare se citesc din fișierul `furtuna.in` care conține pe prima linie coordonatele punctului de start, pe a doua linie coordonatele punctului final, iar pe a treia linie predicția pe zile a direcțiilor în care acționează furtuna.

Datele de ieșire se scriu în fișierul `furtuna.out`. Pe fiecare linie se reprezintă în ordine comanda dată navetei PJ (`L`,`R`,`U`,`D`,`S`) și coordonatele unde a ajuns în urma aplicării comenzii, compusă eventual cu acțiunea furtunii.

Se cere să îl ajutați pe Lt. Col. John Sheppard, comandantul navetei PJ, să ia deciziile de navigare corecte astfel încât să ajungă din punctul inițial în punctul final în cât mai puține zile, dat fiind resursele limitate de supraviețuire care se găsesc pe naveta PJ.

Timpul maxim de execuție pentru un test este de 1 secundă.

furtuna.in	furtuna.out
0 0 9 9 UUUUU	R 1 1 R 2 2 R 3 3 R 4 4 R 5 5 R 6 5 R 7 5 R 8 5 R 9 5 U 9 6 U 9 7 U 9 8 U 9 9

Condiții

Când testăm programe cu condiții complexe, dorim să arătăm că orice variabilă din condiție poate afecta rezultatul, adică, dând variabilei valorile adevărat (`T`) și fals (`F`), rezultatul condiției se schimbă (pentru valori bine alese ale celorlalte variabile). De exemplu, în condiția $a + b * c$, unde * (ȘI) e mai prioritar decât + (SAU), luând $a =$ `F`, $c =$ `T`, valoarea lui b determină valoarea rezultatului.

Scrieți un program care pentru o expresie booleană produce numărul minim de *teste* (combinații de valori) așa încât pentru fiecare variabilă există două teste care arată că ea afectează valoarea expresiei.

Programul va citi din fișierul `conditie.in` o expresie booleană cu operatorii + (SAU), * (ȘI), paranteze `()` și caractere de subliniere _. Fiecare apariție a caracterului de subliniere _ reprezintă o variabilă *distinctă*.

Programul va scrie în fișierul `conditie.out` linii cu câte un test (valori `T` sau `F` pentru fiecare variabilă, separate prin spații). Pentru fiecare variabilă, se scrie apoi câte o linie cu două numere separate prin spațiu, reprezentând testele (numerotate de la 1) care arată că variabila afectează expresia.

Indicație: pentru n variabile, numărul minim de teste e $n + 1$.

conditie.in	conditie.out	Explicație:
_*(_+_)*_	T F T T	Test1: T*(F+T)*T=T
	T T F T	Test2: T*(T+F)*T=T
	F F T T	Test3: F*(F+T)*T=F
	T F F T	Test4: T*(F+F)*T=F
	T F T F	Test5: T*(F+T)*F=F
	1 3	Var1: Test1, Test3
	2 4	Var2: Test2, Test4
	1 4	Var3: Test1, Test4
	1 5	Var4: Test1, Test5

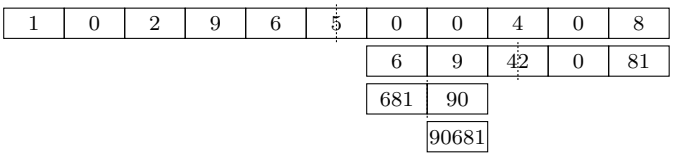
Numere

Ionel este elev în clasa I și, de bucurie că a învățat cifrele, inventează un nou joc. El ia o bandă de hârtie, o împarte în N căsuțe egale și scrie câte o cifră în fiecare căsuță. Pe urmă îndoaie banda în două, exact la jumătate, astfel încât o jumătate din bandă se află sub cealaltă jumătate. Căsuțele sunt aliniate astfel încât fiecare căsuță din jumătatea de sus se află deasupra unei căsuțe din jumătatea de jos. Dacă banda are un număr impar de căsuțe, la mijloc rămâne o căsuță care este ignorată. După o astfel de îndoire, Ionel copiază cifrele din căsuțele aflate dedesupt la sfârșitul cifrelor din căsuțele aflate deasupra (se presupune că are suficient loc în căsuțe), după care scapă de jumătatea de dedesupt a benzii (o desprinde cu foarfeca). El repetă acest procedeu de îndoire și copiere de cifre până când pe bandă rămâne o singură căsuță. Desigur că Ionel știe că cifrele zero aflate la începutul numerelor sunt redundante și le elimină ori de câte ori este posibil.

Determinați dacă există o secvență de îndoiri ale benzii prin care se poate obține în ultima căsuță de pe bandă un anumit număr K. Dacă da, atunci găsiți o astfel de secvență, specificată prin literele `S` (stânga) și `D` (dreapta). Litera `S` înseamnă că jumătatea din stânga a benzii rămâne deasupra, iar litera `D` înseamnă că jumătatea din dreapta rămâne deasupra.

Datele de intrare se citesc din fișierul `numere.in`. Pe prima linie se găsesc numerele N și K. Pe a doua linie se găsesc cele N cifre, separate prin spații. Numărul N nu va fi mai mare de 1000. Datele de ieșire se scriu în fișierul `numere.out`. Dacă există soluție, ea se scrie pe o singură linie, fără spații între litere. Dacă nu există soluție atunci în fișierul de ieșire se scrie textul "`NU`". Dacă există mai multe soluții, găsiți oricare din ele. Timpul maxim de execuție este de 3 secunde/test.

numere.in	numere.out
11 90681 1 0 2 9 6 5 0 0 4 0 8	DSD



numere.in	numere.out
7 431 1 2 3 4 5 6 7	NU