

UFO

PROBLEM

Det er en velkendt hemmelighed, at der bag alle kometer, der nærmer sig jorden, gemmer sig en UFO.

Disse UFO'er kommer for at hente loyale supportere fra jorden. Uheldigvis er pladsen på de enkelte UFO'er begrænset, så man har på forhånd inddelt supporterne i forskellige grupper, som hver har fået tildelt et navn. Grupperne får så besked om, hvem der bliver afhentet næste gang ved hjælp af kometens navn (hvem tror du nemlig, der navngiver kometerne?!).

For at finde ud af dette, skal både kometens og gruppens navn konverteres til et tal efter følgende regel: tallet, der svarer til navnet udregnes ved at gange numrene for de enkelte bogstaver i navnet sammen, idet "A" har nummer 1, "B" nummer 2 og således videre indtil "Z", som har nummer 26 (de bruger det engelske alfabet).

Således svarer gruppen "DATALOG" til $4 \cdot 1 \cdot 20 \cdot 1 \cdot 12 \cdot 15 \cdot 7$, som er lig med 100800, mens kometen "NERDS" får tilordnet tallet 95760.

Hvis gruppens tal modulo 47 er lig med kometens tal modulo 47, skal gruppen holde sig klar til at tage af sted. (Modulo betyder resten ved heltalsdivision således er $94 \text{ modulo } 47 = 0$ og $100800 \text{ modulo } 47 = 32$, mens $95760 \text{ modulo } 47 = 21$).

Opgave

Skriv et program der

- fra standard-input indlæser navnet på kometen og navnet på gruppen.
- beregner om gruppen skal afsted.
- skriver resultatet til standard-output.

INPUT

Input læses fra standard-input og består af to linjer:

- Første linje indeholder en streng af højst 10 store bogstaver. Dette er kometens navn.
- Anden og sidste linje indeholder ligeledes en streng af højst 10 store bogstaver. Dette er gruppens navn.

OUTPUT

Output skal skrives til standard-output og består af en enkelt linje med ordet "GO", hvis gruppen skal afsted, ellers ordet "STAY".

EKSEMPEL

Input

```
DATALOG  
NERDS
```

Output

```
STAY
```

SCORE

Opgaven giver 100 point.

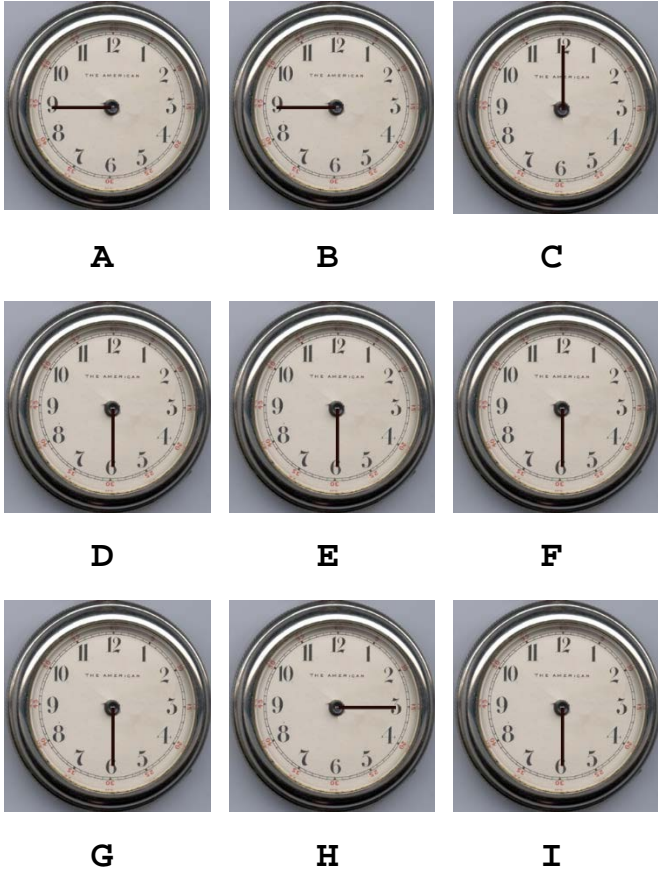
BEGRÆNSNINGER

Tid: 1 sekund

Hukommelse: 256 MB

CLOCKS

PROBLEM



(Figure 1)

There are nine clocks in a 3x3 array (figure 1). The goal is to return all the dials to 12 o'clock with as few moves as possible. There are nine different allowed ways to turn the dials on the clocks. Each such way is called a move. A move will turn the dials 90' (degrees) clockwise on those clocks which are affected according to figure 2 below.

Move	Affected clocks
1	ABDE
2	ABC
3	BCEF
4	ADG
5	BDEFH
6	CFI
7	DEGH
8	GHI
9	EFHI

(Figure 2)

INPUT

Read nine numbers from standard input. These numbers give the start positions of the dials:

- 0 = 12 o'clock ,
- 1 = 3 o'clock ,
- 2 = 6 o'clock ,
- 3 = 9 o'clock .

OUTPUT

Write to standard output a shortest sequence of moves (numbers), which returns all the dials to 12 o'clock. In case there are many solutions, output any one of these.

EXAMPLE

INPUT

```
3 3 0
2 2 2
2 1 2
```

OUTPUT

```
5 8 4 9
```

EXPLANATION

Each number represents a time according to the table above:

3 3 0		3 0 0		3 0 0
2 2 2	5->	3 3 3	8->	3 3 3
2 1 2		2 2 2		3 3 3
		0 0 0		0 0 0
	4 ->	0 3 3	9->	0 0 0
		0 3 3		0 0 0

SCORE

The problem gives 100 points.

LIMITATIONS

Time: 1 second

Memory: 256 MB

SUPERMARKED**PROBLEM**

Mirko er blevet ansat i et nyt super-supermarked, hvor idéen er alting skal være nemmere og hurtigere. Derfor har de valgt at investere i rigtig mange kasser, således at man aldrig behøves at stå i kø ved kassen.

Desværre havde de ikke tænkt over, at når en kunde kommer hen til en kasse, og finder ud af at den er optaget, kan det være svært at finde hen til den nærmeste ledige kasse. Mirko har derfor fået til opgave, at lave et system, der fortæller kunden hvor den nærmeste ledige kasse er (butikken har heldigvis investeret i så mange kasser, at der altid er mindst én ledig kasse, når der kommer en kunde til en kasse). Hvis to kasser er lige tæt på vælges den med laveste nummer. Ligesom i almindelige butikker, er kasserne placeret på en lang række og er nummeret 1, 2, ..., N.

Systemet får besked, når en kunde ankommer til en kasse, og igen når en kunde forlader en kasse. Til at starte med er alle kasser ledige. Hvis en kunde kommer til en ledig kasse, skal systemet blot angive den kasse.

Eksempel. Antag at $N = 5$, og at kasserne 3 og 4 allerede er optaget. Nu ankommer en kunde til kasse 3, som altså er optaget. Systemet fortæller derfor at kunden skal gå til kasse 2, der er den nærmeste ledige (nu er kasserne 2, 3 og 4 optaget). En ny kunde ankommer nu ved kasse 3, da der er lige langt til kasse 1 og 5 fra kasse 3, bestemmer systemet at kunden skal gå til kasse 1.

Kan du hjælpe Mirko med at lave et effektivt system?

INPUT

Input læses fra standard-input og består af flg.:

- Første linje indeholder de to tal N og M (hhv. antal kasser og antal ankomster/afgange)
- De næste M linjer indeholder enten ($1 \leq X \leq N$):
 - "C X": En kunde ankommer til kasse X
 - "L X": En kunde forlader kasse X .

OUTPUT

Output skal skrives til standard-output. For hver gang en kunde ankommer skal der udskrives en linje med et heltal svarende til den kasse kunden skal gå til.

EKSEMPEL INPUT/OUTPUT

Input

```
5 6
C 3
C 5
C 3
C 2
L 3
C 2
```

Output

```
3
5
2
1
3
```

SCORE

Del 1 (50 point): $1 \leq N, M \leq 5.000$

Del 2 (50 point): $1 \leq N, M \leq 1.000.000$

BEGRÆNSNINGER

Tid: 1 sekund

Hukommelse: 256 MB