

# Concours de Programmation ENEAM 2013

## C/C++/JAVA

### Problème A : Prison Break

Michel Scofield a acheté un téléphone par satellite pour l'utiliser dans son évasion. Il doit vérifier que durant sa cavale il sera toujours sous la couverture du satellite auquel il est rattaché. Scofield a noté les points importants où il doit effectuer ses communications avec sa chère Sara Kancredit et avec ses complices et il veut vérifier que ces emplacements seront en visibilité directe avec son satellite de rattachement.

Votre tâche est d'écrire pour Scofield le programme de vérification sachant que la terre est parfaitement sphérique avec un diamètre de 6378km et que le satellite est assimilé à un point.

### Description du format d'entrée

L'entrée contient plusieurs cas de test. La première ligne de chaque test contient le nombre d'emplacement  $n$  à vérifier suivi par la position du satellite : sa latitude (en degrés), sa longitude (en degrés) et son altitude (en km). Chacune des lignes suivantes contient le nom de l'emplacement suivi de son altitude et longitude. Le nom ne dépassera pas une longueur de 60 caractères. Le fichier d'entrée se terminera par  $n = 0$ .

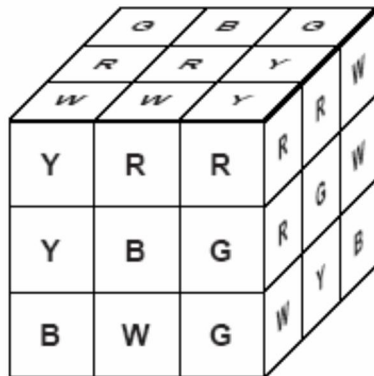
### Description du format de sortie

Pour chaque test, imprimer le numéro du test (voir l'exemple de sortie) suivi du nom des emplacements dont la visibilité est directe avec le satellite défini. Ces emplacements seront imprimés dans leur ordre d'apparition dans le fichier d'entrée. Une ligne vide doit être imprimée après chaque cas de test.

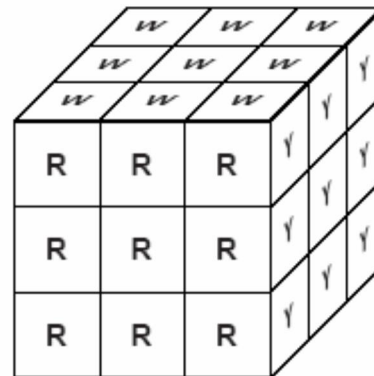
Exemple d'entrée	Exemple de sortie
3 20.0 -60.0 150000000.0 ENEAM 48.406 10.002 NIAMEY -6.13 106.75 HOUENOUSSOU 21.32 -157.83 2 48.4 10 0.5 ENEAM 48.406 10.002 HOUENOUSSOU 21.32 -157.83 0 0.0 0.0 0.0	Test case 1: ENEAM HOUENOUSSOU Test case 2: ENEAM

## Problème B : Cube

Tout le monde connaît le fameux cube avec six couleurs, où le but du jeu est de faire des rotations des faces du cube pour obtenir un cube avec des faces homogènes.

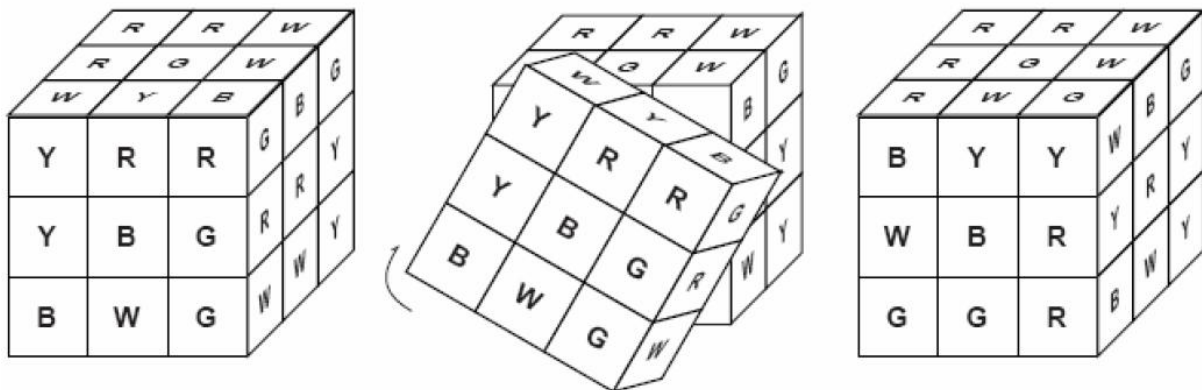


Etat initial



Etat final correct

Une rotation d'une face du cube se fait de la façon suivante :



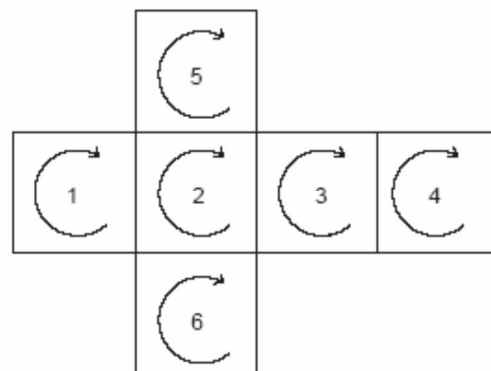
Le but de ce problème est, ayant l'état initial du cube et une série de rotation, de déterminer si le résultat final est correct ou pas.

Pour modéliser le problème, on utilise le schéma suivant :

```

W R W
W W M
G W M
Y Y B R M G W G G R B R
Y Y M B M M Y G G W R R
M G R Y Y M B G M Y W B
B B Y
B B R
G R W

```



Chaque lettre majuscule présente une couleur, les rotations seront présentées par des entiers entre -6 et 6 avec une valeur positive pour les rotations dans le sens de l'horloge et une valeur négative pour une rotation dans le sens contraire.

Spécification de l'entrée :

Le fichier d'entrée contient plusieurs cas, la première ligne contient le nombre des cas. Chaque cas contient dix lignes, les 9 premières contiennent l'état initial du cube. La dixième ligne contient la série des rotations avec 0 à la fin.

Spécification de la sortie :

Pour chaque cas vous devez générer une ligne avec la valeur « YES » si la série de rotation donne un résultat correct et « NO » dans le cas contraire.

Aucune ligne vide supplémentaire à la fin ou au début du fichier ne doit figurer.

Exemple d'entrée/sortie :

Exemple d'entrée	Exemple de sortie
3 G Y Y G Y Y G Y Y W W W Y R R M M M G G B W W W Y R R M M M G G B W W W Y R R M M M G G B R B B R B B R B B -1 0 G Y Y G Y Y G Y Y W W W Y R R M M M G G B W M W Y R R M W M G G B W W W Y R R M M M G G B R B B R B B R B B -1 0 M W M W W G W W Y G Y Y M M B M B G W R B B Y Y M M B M G G W R R Y M G W B B R R G R R W	YES NO YES

R Y Y G B Y R G B +4 +6 -2 +3 -4 +2 -3 -6 0	
--	--

## Problème C : Décodage

Le cryptage de César est l'une des premières méthodes de cryptage. Le principe de ce cryptage est très simple il consiste à changer chaque lettre par une autre lettre comme suit :

Texte crypté :

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Texte clair :

V W X Y Z A B C D E F G H I J K L M N O P Q R S T U

Les autres caractères non alphabétiques restent inchangés. Le but de ce problème est de décoder les phrases cryptées par la méthode de César.

Spécification de l'entrée :

Le fichier d'entrée contient plusieurs cas, chaque cas contient 3 lignes : La première ligne contient le mot « START ». La deuxième ligne contient une ligne avec des mots cryptés. La troisième ligne contient le mot « END ». Le fichier se termine par une ligne contenant le mot « ENDOFINPUT ».

Spécification de la sortie :

Pour chaque cas vous devez générer une ligne contenant la phrase décryptée. Aucune ligne vide supplémentaire à la fin ou au début du fichier ne doit figurer.

Exemple d'entrée standard :

```
START
NS BFW, JAJSYX TK NRUTYFSHJ FWJ YMJ WJXZQT TK YWNANFQ HFZXJX
END
START
N BTZQI WFYMJW GJ KNWXY NS F QNYOQJ NGJWNFS ANQQFLJ YMFS XJHTSI NS WTRJ
END
START
IFSLJW PSTBX KZQQ BJQQ YMFY HFJXFW NX RTWJ IFSLJWTZX YMFS MJ
END
ENDOFINPUT
```

Exemple de sortie standard :

```
IN WAR, EVENTS OF IMPORTANCE ARE THE RESULT OF TRIVIAL CAUSES
I WOULD RATHER BE FIRST IN A LITTLE IBERIAN VILLAGE THAN SECOND IN ROME
DANGER KNOWS FULL WELL THAT CAESAR IS MORE DANGEROUS THAN HE
```

## Problème D : Sèna

Sèna est un génie en Mathématiques, mais il a un petit problème c'est qu'il ne sait pas faire l'addition et la soustraction une fois qu'il y a les parenthèses. Votre mission est de l'aider à le faire. Les opérations à calculer sont de la forme suivante :

$\text{expr} ::= \text{term} \mid \text{expr '+' term} \mid \text{expr '-' term}$

$\text{term} ::= \text{'(' expr ')'} \mid \text{'(' unary_op ')'} \mid \text{literal}$

$\text{unary\_op} ::= \text{'+' term} \mid \text{'-' term}$

$\text{literal} ::= [0-9]$

### Spécification de l'entrée :

Le fichier d'entrée contient plusieurs lignes, chaque ligne contient une expression à calculer. Il n'y a pas de ligne vide à la fin du fichier.

### Spécification de la sortie :

Pour chaque ligne du fichier d'entrée vous devez générer une ligne qui contient le résultat de l'expression. Aucune ligne vide supplémentaire à la fin ou au début du fichier ne doit figurer.

Exemple d'entrée	Exemple de sortie
1	1
(-2)+3	1
(1-(2+3))	-4
(1-2+3)	2
(1-(+(2-3)))	2