



Concours National de Programmation Bénin 08 mars 2008 C, C++ et JAVA

Problème A : Factoriel trié (ballon blanc)

Le factoriel d'un nombre strictement positif s'obtient en multipliant tous les nombres allant de 1 jusqu'à ce nombre. Par exemple le factoriel de 4 s'obtient en faisant $1 \times 2 \times 3 \times 4$ ce qui donne 24 ; le factoriel de 5 s'obtient en faisant $1 \times 2 \times 3 \times 4 \times 5$ ce qui donne 120.

Votre tâche consiste, étant donné un nombre N, à trouver le plus petit nombre qui peut être obtenu en utilisant l'ensemble des chiffres contenus dans le factoriel de N.

Entrée Standard

L'entrée commence avec un seul entier positif N sur la première ligne, indiquant le nombre d'entiers qui vont suivre. Il est suivi par N entiers, chacun sur une seule ligne. Chacun de ces entiers est inférieur ou égal à 100.

Sortie Standard

Pour chacun des N entiers, affichez le plus petit nombre qui peut être obtenu en utilisant l'ensemble des chiffres contenus dans le factoriel de chaque entier, suivant le format indiqué dans l'exemple, c'est-à-dire un résultat par ligne.

Exemple d'entrée

5
4
5
7
10
50

Exemple de sortie correspondante

24
12
45
23688
1111111222333333444444455666666666777778888899

Problème B : Prison Break (ballon bleu)

Une prison dispose d'un grand hall contenant n cellules, l'une à côté de l'autre. Chaque cellule contient un prisonnier, et chaque cellule est fermée.

Une nuit, le gardien de prison s'ennuie et décide de jouer un jeu. Pour le round 1 du jeu, il prend un verre de whiskey, puis coure dans le hall pour ouvrir toutes les cellules. Pour le round 2 du jeu, il prend un verre de whiskey, puis coure dans le hall pour fermer toute autre cellule (cellules 2, 4, 6, ...). Pour le round 3 du jeu, il prend un verre de whiskey, puis coure dans le hall. Il visite chaque troisième cellule (cellules 3, 6, 9, ...). Si la cellule est fermée, il l'ouvre ; si elle est ouverte, il la ferme. Il répète ceci jusqu'à n rounds, prend un dernier verre et sort.

Un certain nombre de prisonniers, peut être zéro, se rendent compte que leurs cellules sont ouvertes et que le gardien est ivre. Ils s'échappent immédiatement.

Etant donné le nombre de cellules, votre tâche est de déterminer combien de prisonniers s'échappent de la prison.

Entrée Standard

L'entrée commence avec un seul entier positif M sur la première ligne, indiquant le nombre d'entiers qui vont suivre. Il est suivi par M entiers compris entre 5 et 100 inclus, chacun sur une seule ligne désignant le nombre de cellules n .

Sortie Standard

Pour chacun des M entiers n , vous devez afficher le nombre de prisonniers qui s'échappent, suivant le format indiqué dans l'exemple, c'est-à-dire un résultat par ligne.

Exemple d'entrée

```
2
5
100
```

Exemple de sortie correspondante

```
2
10
```

Problème C : Colore moi moins (ballon vert)

Le département « Logiciels » de MapCom Technologies développe un logiciel de dessin qui utilise la méthode de réduction des couleurs. Cette méthode procède par analogie. En fait, le logiciel choisit, sur l'échelle de couleurs, celle qui ressemble le plus à chaque couleur de base, et fait ainsi une sorte de moyenne. Il existe trois couleurs de base R (Rouge) G (Green ou Verte) et B (Bleue).

L'entrée consiste en un ensemble cible de seize valeurs de couleurs RGB, et une collection de couleurs arbitraires RGB à associer à la couleur la plus proche contenue dans l'ensemble cible. Dans ce problème, une couleur RGB est définie comme un triplé ordonné (R, G, B) où chaque valeur du triplet est un entier compris entre 0 et 255. La distance entre deux couleurs est définie comme la distance Euclidienne entre deux points tri-dimensionnels. C'est-à-dire, étant donné deux couleurs (R_1, G_1, B_1) et (R_2, G_2, B_2) , leur distance D est donnée par l'équation :

$$D = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

Entrée Standard

L'entrée est une liste de couleurs RGB, une par ligne, spécifiée par trois entiers compris entre 0 et 255, délimités par un seul espace. Les seize premières couleurs forment l'ensemble cible des couleurs auxquelles les couleurs restantes doivent être associées. L'entrée est terminée par une ligne contenant trois valeurs '-1'.

Sortie Standard

Pour chaque couleur à associer, afficher la couleur et la couleur la plus proche contenue dans l'ensemble cible, suivant le format indiqué dans l'exemple ci-dessous.

Exemple d'entrée

```
0 0 0
255 255 255
0 0 1
1 1 1
128 0 0
0 128 0
128 128 0
0 0 128
126 168 9
35 86 34
133 41 193
```

```

128 0 128
0 128 128
128 128 128
255 0 0
0 1 0
0 0 0
255 255 255
253 254 255
77 79 134
81 218 0
-1 -1 -1

```

Exemple de sortie correspondante

```

(0,0,0) est associe a (0,0,0)
(255,255,255) est associe a (255,255,255)
(253,254,255) est associe a (255,255,255)
(77,79,134) est associe a (128,128,128)
(81,218,0) est associe a (126,168,9)

```

Problème D : Renverser les crêpes brûlées (ballon jaune)

Le cuisinier de la « Crêperie Magique de Cotonou » somnole souvent pendant la cuisson des crêpes. Comme conséquence, un côté d'une pile de crêpes est souvent brûlé. Il est clair que servir des crêpes visiblement brûlées aux patrons est une mauvaise affaire. Avant de servir, la serveuse arrangera les piles de crêpes de sorte que les côtés brûlés fassent face vers le bas.

Vous devez écrire un programme pour aider la serveuse à empiler les crêpes correctement.

Nous commençons par une pile de N crêpes de tailles distinctes, dont chacune est brûlée d'un côté. Le problème est de convertir la pile de crêpes en une pile dans laquelle les crêpes sont dans l'ordre de taille avec la plus petite crêpe en haut de la pile et la plus grande en bas de la pile et le côté brûlé vers le bas pour chaque crêpe. Pour faire ceci, on nous permet de renverser les k crêpes du dessus en bloc (ainsi la k-ième crêpe est maintenant en haut de la pile et la crêpe précédemment en haut est maintenant dans la k-ième position et le côté brûlé va de haut en bas et vice versa).

Par exemple (+ indique le bas brûlé, - un dessus brûlé) :

$+1 \ -3 \ -2$ [renverse 2] $\Rightarrow +3 \ -1 \ -2$ [renverse 1] $\Rightarrow -3 \ -1 \ -2$ [renverse 3] \Rightarrow

$+2 \ +1 \ +3$ [renverse 1] $\Rightarrow -2 \ +1 \ +3$ [renverse 2] $\Rightarrow -1 \ +2 \ +3$ [renverse 1] $\Rightarrow +1 \ +2 \ +3$

Vous devez écrire un programme qui trouve une séquence de tout au plus $(3N - 2)$ renversements, qui convertit une pile donnée de crêpes en une pile assortie avec tous les côtés brûlés vers le bas.

Entrée Standard

La première ligne de l'entrée contient un seul nombre entier décimal, N désignant le nombre de cas qui suivront. Chacune des N lignes suivantes donne un cas d'essai différent représentant une séquence de nombres séparés par des espaces. Le premier nombre sur chaque ligne donne le nombre M de crêpes dans le cas d'essai. Le reste du cas d'essai est les nombres 1 à M dans un certain ordre, chacun avec un signe plus ou moins, donnant la pile initiale de crêpes. Les nombres indiquent les tailles relatives des crêpes et les signes indiquent si le côté brûlé est en haut (-) ou en bas (+). M sera inférieur ou égal à 30.



Concours National de Programmation Bénin 08 mars 2008 C, C++ et JAVA

Sortie Standard

Pour chaque cas d'essai, vous devez générer une ligne de sortie avec les valeurs suivantes : le numéro du cas d'essai sous la forme d'un entier décimal (commençant par un), un espace, le nombre de renversements (K où $K \geq 0$) nécessaires pour arranger les crêpes et la séquence des K nombres, donnant chacun le nombre de crêpes à renverser (dans l'ordre de renversement).

Il peut exister plusieurs solutions correctes pour un cas d'essai. Par exemple 3 2 3 est aussi une solution au premier cas d'essai de l'exemple ci-dessous.

Exemple d'entrée

```
3
3 +1 -3 -2
4 -3 +1 -2 -4
5 +1 +2 +3 +4 -5
```

Exemple de sortie correspondante

```
1 6 2 1 3 1 2 1
2 6 4 1 4 3 1 2
3 3 5 1 5
```



Concours National de Programmation Bénin 08 mars 2008 C, C++ et JAVA

Problème E : Score du concours national (ballon marron)

MapCom Technologies organise le Concours National de Programmation du Bénin. A l'issue de ce concours, les 3 premières équipes seront récompensées. Les meilleures équipes encore inscrites dans des universités l'année prochaine seront prises en charge pour représenter notre pays au Concours Régional de Programmation ACM/ICPC sponsorisé par IBM.

Dans le Concours National de Programmation, les solutions soumises aux juges s'appellent des « runs ». Chaque « run » est soit accepté, soit rejeté par les juges et l'équipe est notifiée du résultat.

Les équipes sont classées suivant le plus grand nombre de problèmes résolus. Pour les besoins de la cérémonie de remise de prix, les équipes qui résolvent le même nombre de problèmes sont classées selon le moins de temps total utilisé et pénalisé. Le temps total utilisé et pénalisé est la somme du temps consommé pour chaque problème résolu. Le temps consommé pour chaque problème résolu est le temps qui s'écoule depuis le début du concours jusqu'à la soumission du « run accepté » plus vingt (20) minutes de pénalités pour chaque « run rejeté ». Il n'y a pas de temps consommé pour un problème non résolu.

Votre programme doit lire une liste de jugements des « runs » et afficher les scores pour les 1^{ère}, 2^{ème} et 3^{ème} équipes.

Entrée Standard

L'entrée peut contenir plusieurs cas d'essais. Chaque cas d'essai est constitué de deux parties. La première partie est une seule ligne contenant un nombre positif (supérieur ou égal à 3) qui indique le nombre d'équipes. Les lignes qui suivent sont les jugements des « runs ». Il y a une ligne par « run », indiquant le temps de soumission (supérieur à 0), le numéro de l'équipe (compris entre 1 et 99 inclus), le numéro du problème (compris entre 1 et 8 inclus) et le jugement (oui ou non). La fin du cas est indiquée par un zéro sur une seule ligne.

La dernière ligne de l'entrée contient seulement zéro comme nombre d'équipes. Ce cas ne doit bien sûr pas être traité.

Sortie Standard

Pour chaque cas d'essai, vous devez générer le score des trois premières équipes, conformément au format indiqué dans l'exemple ci-dessous. Le score contient le numéro de l'équipe, le rang, le nombre de problèmes résolus, le temps total utilisé et pénalisé.



Concours National de Programmation Bénin 08 mars 2008 C, C++ et JAVA

NB : Plusieurs équipes peuvent occuper la même place. Par exemple, si plusieurs équipes sont 3^{ème}, affichez les toutes suivant l'ordre croissant des numéros. Si plusieurs équipes sont en 2^{ème} place, elles seront toutes affichées suivant l'ordre croissant des numéros et alors il n'y aura pas de 3^{ème} équipe à afficher ou à récompenser, etc.

Exemple d'entrée

```
3
12 1 2 oui
14 3 2 non
25 3 1 oui
29 1 1 non
38 3 2 oui
39 2 1 non
45 1 1 non
0
4
0
5
12 1 2 oui
12 2 1 oui
20 3 4 non
20 4 5 non
30 3 4 oui
30 4 5 oui
0
0
```

Exemple de sortie correspondante

```
Cas #1:
No. Equipe  Rang  Problemes resolus  Temps consomme
          3      1              2             83
          1      2              1             12
          2      3              0              0

Cas #2:
No. Equipe  Rang  Problemes resolus  Temps consomme
          1      1              0              0
          2      1              0              0
          3      1              0              0
          4      1              0              0

Cas #3:
No. Equipe  Rang  Problemes resolus  Temps consomme
          1      1              1             12
          2      1              1             12
          3      3              1             50
          4      3              1             50
```


Problème F : Tourne, tourne et tourne encore ! (ballon gris)

Un nombre cyclique est un entier composé de n chiffres et qui multiplié par n'importe quel entier allant de 1 à n donne un « cycle » des chiffres du nombre original. C'est-à-dire, si vous considérez le nombre après que le dernier chiffre ait été « enroulé » au premier chiffre, l'ordre des chiffres dans les deux nombres sera identique, bien qu'ils puissent commencer à différentes positions.

Par exemple, le nombre 142857 est cyclique, comme illustré ici :

$$142857 \times 1 = 142857$$

$$142857 \times 2 = 285714$$

$$142857 \times 3 = 428571$$

$$142857 \times 4 = 571428$$

$$142857 \times 5 = 714285$$

$$142857 \times 6 = 857142$$

Vous devez écrire un programme qui détermine si les nombres sont cycliques ou non.

Entrée Standard

L'entrée commence avec un seul entier positif M sur la première ligne, indiquant le nombre d'entiers qui vont suivre. Il est suivi par M entiers, chacun sur une seule ligne et ayant chacun un nombre de chiffres compris entre 2 et 60 inclus. (Notez que des zéros débutant le nombre ne devraient pas être enlevés, ils sont considérés comme une partie du nombre et doivent être pris en compte dans la détermination de n . Ainsi, « 01 » est un nombre de deux chiffres, distinct de « 1 » qui est un nombre d'un chiffre).

Sortie Standard

Pour chacun des M entiers de l'entrée, vous devez écrire une ligne qui indique s'il est cyclique ou non en suivant le format spécifié dans l'exemple ci-dessous.

Exemple d'entrée

```
5
142857
142856
142858
```



01
0588235294117647

Exemple de sortie correspondante

142857 est cyclique
142856 n'est pas cyclique
142858 n'est pas cyclique
01 n'est pas cyclique
0588235294117647 est cyclique

Problème G : Anagrammes (ballon orange)

Deux mots sont des anagrammes l'un pour l'autre si l'un peut être obtenu en permutant certaines lettres de l'autre mot. C'est le cas de « proie » et de « poire » qui sont des anagrammes l'un pour l'autre. (Notez que « proie » et « proiee » ne sont pas des anagrammes l'un pour l'autre étant donné que l'un contient plus de lettres que l'autre)
Vous développez une partie d'un logiciel du jeu Scrabble et vous devez tester si deux mots sont des anagrammes l'un pour l'autre.

Dans ce problème, tout mot est un anagramme pour lui-même et un mot est composé uniquement de lettres minuscules choisies parmi les vingt six lettres de l'alphabet « abcdefghijklmnopqrstuvwxyz ». Par exemple, « cnpbenin » est un mot. Le mot le plus long est « bienvenueauconcoursnationaldeprogrammationbeninlevenementdelannee ».

Entrée Standard

L'entrée commence avec un seul entier positif N sur la première ligne, indiquant le nombre de cas qui vont suivre. Il est suivi par N cas, chacun sur une seule ligne. Chaque cas comprend deux mots séparés par un seul espace.

Sortie Standard

Pour chacun des N cas l'entrée, vous devez écrire une seule ligne qui indique si les deux mots sont des anagrammes l'un pour l'autre, en suivant le format spécifié dans l'exemple ci-dessous.

Exemple d'entrée

```
4
poire proie
anagramm anagramme
cafe face
lokossa parakou
```

Exemple de sortie correspondante

```
poire et proie sont des anagrammes l'un pour l'autre
anagramm et anagramme ne sont pas des anagrammes l'un pour l'autre
cafe et face sont des anagrammes l'un pour l'autre
lokossa et parakou ne sont pas des anagrammes l'un pour l'autre
```

Problème H : Distance anagrammatique (ballon rouge)

Deux mots sont des anagrammes l'un pour l'autre si l'un peut être obtenu en permutant certaines lettres de l'autre mot. C'est le cas de « proie » et de « poire » qui sont des anagrammes l'un pour l'autre. (Notez que « proie » et « proiee » ne sont pas des anagrammes l'un pour l'autre étant donné que l'un contient plus de lettres que l'autre)

La distance anagrammatique entre deux mots est le nombre minimal de lettres qu'il faut enlever pour obtenir deux mots anagrammes l'un pour l'autre. Par exemple, la distance anagrammatique entre « prendre » et « dent » est 5 puisqu'il faut enlever la lettre *p*, deux lettres *r* et une lettre *e* du mot « prendre » et la lettre *t* du mot « dent » pour obtenir deux mots « end » et « den » qui sont anagrammes l'un pour l'autre.

Dans ce problème, tout mot est un anagramme pour lui-même et un mot est composé uniquement de lettres minuscules choisies parmi les vingt six lettres de l'alphabet « abcdefghijklmnopqrstuvwxyz ». Par exemple, cnpbenin est un mot. Le mot le plus long est « bienvenueauconcoursnationaldeprogrammationbeninlevenementdelannee ».

Entrée Standard

L'entrée commence avec un seul entier positif *N* sur la première ligne, indiquant le nombre de cas qui vont suivre. Il est suivi par *N* cas. Chaque cas comprend deux mots, chacun sur une seule ligne.

Sortie Standard

Pour chacun des *N* cas l'entrée, vous devez écrire une seule ligne qui indique la distance anagrammatique entre les deux mots, en suivant le format spécifié dans l'exemple ci-dessous.

Exemple d'entrée

```
5
poire
proie
anagramm
anagramme
cafe
face
lokossa
parakou
changement
emergence
```



Exemple de sortie correspondante

Cas #1: 0
Cas #2: 1
Cas #3: 0
Cas #4: 8
Cas #5: 7