

Lëtzebuerger Informatiksolympiad 2014

Qualifikationsronn

Problème I

Super integer

Tâche

La factorielle d'un nombre entier positif ou nul est définie de la manière suivante:

$$0! = 1$$

$$n! = n \cdot (n - 1)!$$

Exemple:

$$4! = 4 \cdot 3! = 4 \cdot 3 \cdot 2! = 4 \cdot 3 \cdot 2 \cdot 1! = 4 \cdot 3 \cdot 2 \cdot 1 \cdot 0! = 4 \cdot 3 \cdot 2 \cdot 1 \cdot 1 = 24$$

Il s'agit d'écrire un programme qui calcule la factorielle d'un nombre entier N entré au clavier et qui affiche la somme des chiffres le composant. Dans l'exemple ci-dessus, la somme des chiffres de $4!$ vaut $2 + 4 = 6$.

Restriction

N est un entier tel que $0 \leq N \leq 500$.

Entrée et sortie du programme

Entrée: Le programme lit au clavier le nombre N .

Sortie: Le programme affiche la somme des chiffres composant le nombre $N!$

Exemples d'exécution

Si N est égal à 10, l'affichage à la console sera:

10

27

Si N est égal à 500, l'affichage à la console sera:

500

4599

Remettez le programme sous le nom FACTORIELLE.xxx, avec xxx=PAS ou C(PP) ou JAVA.

Problème II

Triangles rectangles

Tâche

Il s'agit d'écrire un programme qui détermine pour quelle plus grande valeur entière $P \leq N$ du périmètre d'un triangle rectangle il existe un maximum de triangles rectangles à côtés entiers.

Si $N = 35$ par exemple, il existe trois périmètres entiers d'un triangle rectangle à savoir 12, 24 et 30 dont les côtés mesurent respectivement (3, 4, 5), (6, 8, 10) et (5, 12, 13) et dans ce cas la réponse est $P = 30$.

Restriction

N est un entier tel que $1 \leq N \leq 2000$.

Entrée et sortie du programme

Entrée: Le programme lit au clavier le nombre N .

Sortie: Le programme affiche la plus grande valeur de $P \leq N$ possible tel que pour cette valeur de P , il existe un maximum de triangles rectangles à côtés entiers ainsi que le nombre de triangles.

Exemples d'exécution

Si N est égal à 2000, l'affichage à la console sera:

```
2000
Périmètre solution: 1680
Nombre de triangles: 10
```

Si N est égal à 100, l'affichage à la console sera:

```
100
Périmètre solution: 90
Nombre de triangles: 2
```

Si N est égal à 10, l'affichage à la console sera:

```
10
Périmètre solution: 0
Nombre de triangles: 0
```

Remettez le programme sous le nom TRIANGLES.xxx, avec xxx=PAS ou (PP) ou JAVA.

Problème III

Messages cryptés

Tâche

Un groupe de jeunes gens préfère crypter les messages qu'ils s'envoient mutuellement. Ils se basent exclusivement sur les lignes ci-dessous du code ASCII standard, présentées comme suit:

| | | | | | | | | | | | | | | | |
|----|---|----|---|----|---|----|---|----|---|----|---|----|---|----|---|
| 64 | @ | 65 | A | 66 | B | 67 | C | 68 | D | 69 | E | 70 | F | 71 | G |
| 72 | H | 73 | I | 74 | J | 75 | K | 76 | L | 77 | M | 78 | N | 79 | O |
| 80 | P | 81 | Q | 82 | R | 83 | S | 84 | T | 85 | U | 86 | V | 87 | W |
| 88 | X | 89 | Y | 90 | Z | 91 | [| 92 | \ | 93 |] | 94 | ^ | 95 | _ |

Devant le corps du message ils mettent la **clé**, composée d'exactly quatre caractères choisis parmi ceux du tableau ci-dessus. Chaque caractère de la clé indique le décalage à appliquer pour l'encryptage d'un caractère de la ligne concernée du tableau ci-dessus: le premier caractère de la clé concerne les caractères alphabétiques de la première ligne (celle commençant par le caractère de code 64), le deuxième caractère de la clé les caractères alphabétiques de la deuxième ligne (celle commençant par le caractère de code 72), etc. Les quatre caractères de la clé ne sont pas encryptés, mais transmis tels quels.

L'encryptage des caractères qui forment le **corps du message** fonctionne par décalage au sein de la même ligne du tableau indiqué ci-dessus. Dès que le décalage d'un caractère donnerait lieu au dépassement de la ligne, on revient au début de la ligne. Chaque caractère de la clé provoque un décalage correspondant à son éloignement du début de la ligne dans laquelle il se trouve.

L'illustration qui suit devrait permettre de comprendre. Si par exemple les quatre caractères de la clé sont YB@_ et étant donné que:

- Y est le premier caractère indiqué, on s'en sert pour encrypter les caractères de la première ligne du tableau et comme il est éloigné d'une position du début de sa ligne (occupe la deuxième position dans sa ligne) dans la table ASCII, un caractère entre A et G est encrypté par un décalage de 1 dans la ligne correspondante (A devient B, B devient C, ..., G devient @),
- B est le deuxième caractère indiqué, on s'en sert pour encrypter les caractères de la deuxième ligne du tableau et comme il est éloigné de deux positions du début de sa ligne (occupe la troisième position dans sa ligne) dans la table ASCII, un caractère entre H et O

est encrypté par un décalage de 2 dans la ligne correspondante (H devient J, ..., N devient H, O devient I),

- @ est le troisième caractère indiqué, on s'en sert pour encrypter les caractères de la troisième ligne du tableau et comme il occupe la première position dans sa ligne dans la table ASCII, un caractère entre P et W est encrypté sans décalage (P reste P, Q reste Q, ..., W reste W),
- _ est le quatrième caractère indiqué, on s'en sert pour encrypter les caractères de la quatrième ligne du tableau et comme il est éloigné de sept positions du début de sa ligne (occupe la dernière position dans sa ligne) dans la table ASCII, un caractère entre X et Z est encrypté par un décalage de 7 dans la ligne correspondante (X devient _, Y devient X, ..., Z devient Y).

Il s'agit d'écrire un programme qui permet d'encrypter et de décrypter un tel message.

Restrictions

Tous les caractères du tableau ci-dessus peuvent faire partie de la clé. Le corps du message proprement dit comprend au maximum 250 caractères, considérant que:

- seuls les caractères du tableau ci-dessus sont encryptés ou décryptés,
- le caractère « ! » est considéré comme symbole de fin du message et n'est pas restitué,
- tout autre caractère sera restitué comme un espace.

Entrée du programme

Principe

Une entrée comprendra un maximum de 256 caractères et aura toujours la forme suivante:

<action><clé><corps du message>!

| Élément du message | Explication |
|--------------------|---|
| <action> | + pour encrypter - pour décrypter |
| <clé> | exactement quatre caractères dont le code ASCII se situe entre 64 et 95 |
| <corps du message> | 250 caractères |
| ! | le point d'exclamation termine la saisie et ne compte pas comme caractère du corps du message |

Exemple 1

+YB@_ABCDE FGHIJK LMNOP QRSTUVW XYZ!

Exemple 2

-YB@_BCDEF G@JKLM NOHIP QRSTUVW _XY!

Exemple 3

-VITE G MG VCAICVAIC BQ NCJMMCQV GMEHVJPINC!

Sortie du programme**Principe**

Les quatre premiers caractères entrés sont affichés sans transformation, pour les autres caractères on affiche le caractère qui résulte des règles indiquées ci-dessus. Le point d'exclamation qui termine l'entrée n'est pas affiché.

Les exemples ci-dessous résultent de ceux donnés ci-dessus dans le cadre de l'entrée.

Exemple 1

YB@_BCDEF G@JKLM NOHIP QRSTUVW _XY

Exemple 2

YB@_ABCDE FGHIJK LMNOP QRSTUVW XYZ

Exemple 3

VITE A LA RECHERCHE DU MEILLEUR ALGORITHME

Remettez le programme sous le nom MESSAGES.xxx, avec xxx=PAS ou C(PP) ou JAVA.

Problème IV**Activités de weekend****Tâche**

Pendant toute l'année la commune X organise des activités de weekend pour les enfants habitant la commune. Afin de permettre à tout enfant d'en profiter pleinement, elle propose toute une liste d'ateliers auxquels les enfants peuvent s'inscrire.

Une fois que les horaires des ateliers ont été fixés et que les enfants ont sélectionné les ateliers auxquels ils souhaitent participer, il s'agit de détecter combien d'enfants ne pourront pas suivre tous les ateliers choisis.

Il s'agit d'écrire un programme qui indique le nombre d'enfants qui ne seront pas en mesure de suivre tous les ateliers choisis.

Restrictions

Il faudra respecter les restrictions suivantes:

- un code d'atelier **CA** est exprimé sous forme de 1 à 8 caractères alphabétiques majuscules non accentués (lettres de A à Z),
- le jour de semaine **J** peut prendre les valeurs S pour Samedi et D pour Dimanche,
- l'heure début **H** d'un atelier est toujours une heure entière et les ateliers peuvent démarrer dès 8 heures et au plus tard à 16 heures: $8 \leq H \leq 16$,
- la durée **D** d'un atelier est d'au moins une et d'au plus 6 heures entières: $1 \leq D \leq 6$,
- le nombre **A** d'ateliers proposés est entre 1 et 100: $1 \leq A \leq 100$,
- le nombre **E** d'enfants inscrits est entre 1 et 1000: $1 \leq E \leq 1000$,
- le nombre **N** d'ateliers choisis par un enfant est compris entre 1 et 5: $1 \leq N \leq 5$.

Entrée du programme

Principe

On fournit

- le nombre d'ateliers **A** suivi de A lignes comprenant chacune un code d'atelier CA ainsi que le jour J, l'heure début H et la durée en heures entières D, les différentes indications étant séparées par des espaces,
- le nombre d'enfants **E** suivi de E lignes comprenant chacune le nombre d'ateliers choisis N suivi d'un espace et de N codes d'atelier, indiqués dans un ordre quelconque et séparés par des espaces.

Exemple 1

5

SPORT S 10 2

DESSIN D 11 2

CANOE S 11 5

PLONGEE S 15 2

ECHECS D 10 2

3

3 SPORT DESSIN CANOE

2 PLONGEE ECHECS

3 PLONGEE SPORT CANOE

Exemple 2

```
3
SPORT D 10 2
DESSIN D 12 2
CANOE D 14 4
2
3 CANOE SPORT DESSIN
2 SPORT CANOE
```

Exemple 3

```
6
SPORT S 10 2
DESSIN S 11 2
CANOE S 11 5
PLONGEE S 15 2
ECHECS S 10 2
INFOR S 08 4
4
1 INFOR
3 SPORT DESSIN CANOE
2 PLONGEE ECHECS
2 PLONGEE SPORT
```

Sortie du programme**Principe**

On affiche le texte « Enfants ne pouvant pas exercer tous les ateliers choisis: » suivi d'un espace et du nombre d'enfants pour lesquels il existe une impossibilité à participer à toutes les activités choisies.

Les exemples ci-dessous résultent de ceux donnés ci-dessus dans le cadre de l'entrée.

Exemple 1

```
Enfants ne pouvant pas exercer tous les ateliers choisis: 2
```

Exemple 2

```
Enfants ne pouvant pas exercer tous les ateliers choisis: 0
```

Exemple 3

```
Enfants ne pouvant pas exercer tous les ateliers choisis: 1
```

Remettez le programme sous le nom WEEKEND.xxx, avec xxx=PAS ou C(PP) ou JAVA.