

Concours Informatique Luxembourgeois 2012 Epreuve Finale (05/07/2012)

Tâche I - Marathon

50 points

C'est l'époque des marathons: Luxembourg, Bruxelles, Paris, etc. Ce type de manifestation concerne de plus en plus de personnes de tout âge et souvent les plus vieux sont mieux classés que les plus jeunes. Les résultats des concurrents sont en général accessibles en ligne et sont présentés par temps d'arrivée, c'est-à-dire depuis le coureur le plus rapide au plus lent, indépendamment de leur âge.

Le but de ce programme est de déterminer le sous-classement le plus long possible par temps d'arrivée mais dans lequel des coureurs sont classés du plus vieux au plus jeune.



Tâche

Soit un fichier texte de données contenant les résultats de la course. Chaque ligne du fichier se compose d'un numéro de dossard suivi de l'âge du participant suivi du temps qu'il a mis. Le nombre de lignes **NL** n'est pas connu à l'avance, la dernière ligne est signalée par le numéro de dossard -1. Voici un exemple d'un tel fichier:

```
20 31 2:08:43
31 32 2:09:07
3 30 2:11:22
12 39 2:13:10
5 30 2:13:33
11 31 2:14:23
43 26 2:14:23
76 31 2:14:30
-1
```

Écrivez un programme qui détermine la longueur de la plus longue liste de coureurs classée par temps d'arrivée et par âge décroissant, chaque âge n'apparaissant qu'une seule fois dans cette liste. [40 points maximum dépendant du nombre de coureurs]

Affichez ensuite une telle liste. [10 points]

Restrictions

Sous-tâche 1 [20 points]

Pour 20 points, le nombre maximal de coureurs classés ne dépassera pas 200, c.-à-d.

$1 \leq \text{NL} \leq 200$

Le nombre maximal de coureurs classés.

Sous-tâche 2 [20 points]

Pour 20 points, le nombre maximal de coureurs classés ne dépassera pas 15000, c.-à-d.

$1 \leq \text{NL} \leq 15000$

Le nombre maximal de coureurs classés.

Entrée et sortie du programme

Entrée

- Le programme lit d'abord à partir de la console le nom du fichier contenant le classement.
- Ce fichier contient les résultats des **NL** coureurs classés. Pour chaque coureur, le fichier contient une ligne qui se compose d'un numéro de dossard (entier) suivi de l'âge du participant (entier) suivi du temps qu'il a mis sous la forme « h:mm:ss ». Ces trois éléments sont séparés par un ou plusieurs espaces. Le nombre de lignes **NL** n'est pas connu à l'avance, la dernière ligne est signalée par le numéro de dossard -1.

Sortie

- Le programme doit afficher la longueur de la plus longue liste de coureurs classée par temps d'arrivée et par âge décroissant, chaque âge n'apparaissant qu'une seule fois dans cette liste et puis afficher une telle liste (de nombreuses solutions sont possibles, il suffit d'en afficher une seule) sous la forme indiquée dans les exemples d'exécution ci-dessous (numéro de dossard suivi d'un tiret suivi de l'âge).

Exemples d'exécution

Contenu du fichier IN1.TXT:

```
20 31 2:08:43
31 32 2:09:07
3 30 2:11:22
12 39 2:13:10
5 30 2:13:33
11 31 2:14:23
43 26 2:14:23
76 31 2:14:30
-1
```

Saisie à la console: IN1.TXT

Affichage à la console:

```
3
31-32 5-30 43-26
```

Contenu du fichier IN2.TXT:

```
1 31 2:08:43
2 32 2:09:07
3 30 2:11:22
4 39 2:13:10
5 30 2:13:33
6 31 2:14:17
7 26 2:14:23
8 31 2:14:30
9 29 2:15:25
10 30 2:16:37
11 27 2:18:38
12 36 2:19:11
13 30 2:19:13
14 25 2:19:54
15 27 2:20:17
16 24 2:20:23
17 22 2:20:38
18 29 2:20:47
19 30 2:21:40
20 26 2:21:54
-1
```

Saisie à la console: IN2.TXT

Affichage à la console:

```
7
2-32 5-30 9-29 11-27 14-25 16-24 17-22
```

Remettez le programme sous le nom MARATHON.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable MARATHON.EXE correspondant au programme.

Spacer est une jeune entreprise innovante qui désire révolutionner le marché des téléphones portables. Ayant découvert que les usagers mettent beaucoup de temps à frapper les nombreux espaces d'un message SMS, Spacer a développé un téléphone portable sans touche « Espace ». En effet Spacer garantit qu'il est possible d'écrire les mots du message sans séparation (p.ex. « Jesuisentraindejouerunjeu ») et que leur système innovant arrive à les découper et séparer. Pour cela, Spacer utilise un dictionnaire fourni avec le téléphone qui contient tous les mots possibles et imaginables.



Tâche

Écrivez un programme qui, étant donné un message **M** et un dictionnaire de mots **D**, permet de segmenter le message **M** en une séquence de mots du dictionnaire **D** les mots étant séparés par exactement un espace.

Restrictions

$A = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$

Le message **M** est formé exclusivement des lettres minuscules de l'alphabet **A**

$1 \leq N_D \leq 250\,000$ Le nombre maximal de mots dans le dictionnaire **D**

Sous-tâche 1 [25 Points]

Pour 25 points, la longueur maximale du message **M** ne dépassera pas 25, c.-à-d.

$1 \leq L_M \leq 25$ La longueur maximale du message **M**

Sous-tâche 2 [25 Points]

Pour 25 points, la longueur maximale du message **M** ne dépassera pas 1000, c.-à-d.

$1 \leq L_M \leq 1000$ La longueur maximale du message **M**

Entrée et sortie du programme

Entrée

Le dictionnaire **D** est fourni dans le fichier DICT.TXT

La première ligne du fichier texte DICT.TXT contient un seul nombre entier positif: le nombre de mots N_D du dictionnaire **D**.

Les N_D lignes suivantes contiennent les différents mots, un mot par ligne. Les mots sont tous écrits en minuscules et sont formés exclusivement des lettres de l'alphabet **A**.

Le message **M** est fourni dans le fichier IN.TXT. Ce fichier contient une seule chaîne de caractères contenu dans une seule ligne. Le message **M** est écrit en minuscules et est formé exclusivement des lettres de l'alphabet **A**. Le message **M** a une longueur maximale L_M

Sortie

Le programme doit créer un fichier texte OUT.TXT contenant une seule ligne. Cette ligne contiendra la segmentation du message **M** en une séquence de mots, deux mots étant séparés par un espace. Il ne doit y avoir un espace en début de ligne ni à la fin de la ligne. Tous les mots de la séquence doivent figurer dans le dictionnaire **D**.

Dans le cas où il n'est pas possible de segmenter complètement le message **M** à l'aide des mots du dictionnaire **D**, le fichier OUT.TXT doit contenir -1.

Exemple d'exécution

Dans le répertoire « SPACER » vous pouvez trouver plusieurs exemples d'exécution où chaque exemple contient le dictionnaire DICT.TXT, le fichier IN.TXT ainsi que le fichier OUT.TXT correspondant.

Faites surtout attention aux exemples 4, 5 et 6 !

Remettez le programme sous le nom SPACER.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable SPACER.EXE correspondant au programme.