L3/M1, 2005–2006

Algorithmique effective – TD 1
Bases, entrées sorties

{Nicolas.Schabanel,Sylvain.Perifel}@ens-lyon.fr

1 Petit rappel Unix

Commandes de base Unix : `ls`, `cd`, `mkdir`, `less`, `cp`, `mv`, `rm`.
Utiliser un éditeur de votre choix pour taper votre code. Par exemple

```
emacs toto.cpp &
```

(le tdman décline toute responsabilité en cas d'utilisation de `vi`).
Pour compiler :

```
echo "alias G++=\"g++-3.4 -Wall -W -pedantic\"" >> ~/.bashrc
G++ -o toto toto.cpp
```

Pour lancer le programme :

```
./toto
./toto < toto.in
./toto < toto.in > toto.out
./toto < toto.in | less
```

2 Commandes utiles de C++ pour le TD

Entête :

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
```

Commandes utiles :

```
std::cin >> x >> y;
std::cin.peek();
std::cin.get();
std::cout << x << ' ' << y << std::endl;
long lrand48();
int atoi(char *s);
```

3 Au travail

Exercice 1
Entrées et sorties

1. Écrire un programme lisant les entrées au format ci-dessous et renvoyant la sortie qui correspond.
   Entrée : la première ligne consiste en un entier n indiquant le nombre de lignes qui vont suivre. Chacune des lignes suivantes commence par un entier inférieur à 1000 indiquant le nombre d'entrées qui suivent sur la ligne. Celles-ci sont des nombres flottants.
   Sortie : la première ligne consiste en l'entier n. Pour $1 < i \leq n+1$, la ligne i de la sortie consiste en les nombres flottants de la ligne i de l'entrée, dans l'ordre inverse.
   Exemple :

`2`	`2`
`1 1.789`	`1.789`
`3 1.515 0.8 -0.052`	`-0.052 0.8 1.515`

2. *Entrée* : des lignes (éventuellement vides) contenant des entiers non nuls, suivies d'une dernière ligne constituée de 0. Chaque ligne contient au plus 1000 entiers.
   Sortie : pour chaque ligne de l'entrée, à part la dernière contenant 0, indiquer sur une nouvelle ligne le nombre d'entiers (éventuellement nul) qu'elle contient.
   Exemple :

`1815`	`1`
`2001 732`	`2`
`0`	

3. Écrire des programmes générant un fichier de tests aléatoires pour chacun des problèmes précédents. On pourra utiliser
   `m = lrand48() % M;`
   pour obtenir un entier pseudo-aléatoire $0 \leq m < M$.

Exercice 2

On pourra utiliser les priority queues, en ajoutant l'entête
`#include <queue>`
puis en déclarant
`priority_queue<Type> tas ;`

1. Écrire un programme sortant, quand on le lui demande, le maximum de l'ensemble en cours de construction.
   Entrée : la première ligne consiste en un entier n indiquant le nombre de lignes qui vont suivre. Chacune des lignes suivantes est une succession de mots, qui sont soit un entier, soit le mot `max`. Aucune ligne ne commence par `max`, et toutes se terminent par l'entier 0 (qui n'apparaît qu'en fin de ligne).
   Sortie : à chaque occurence du mot `max`, indiquer le maximum des entiers de la ligne situés avant ce mot.
   Exemple :

`2`	
`8 2 max 1 1 23 max 3 max 0`	`8 23 23`
`1 max 5 max 2 0`	`1 5`

2. Pour l'université de Paghai, comme chaque année, on souhaite classer les universités internationales par ordre de mérite. Pour cela, on prend en compte deux critères : les frais d'inscription, et le salaire des professeurs. Une université est meilleure qu'une autre lorsque ses frais d'inscription sont plus élevés, ou alors, s'ils sont égaux, lorsque les professeurs sont mieux payés.
   Entrée : la première ligne est constituée d'un entier indiquant le nombre d'universités. Chacune des lignes suivantes contient le nom d'une nouvelle université, suivie de deux nombres flottants représentant dans l'ordre les frais d'inscription, et le salaire des professeurs.
   Sortie : la liste triée des noms des universités, chacune sur une ligne, la meilleure en premier.
   Exemple :

```
3
Cambridge 11.5 93.2
ENSL -1.271 22.1
Berkeley 11.5 112.4
```

```
Berkeley
Cambridge
ENSL
```

Exercice 3
PC : 110501 ; UVa : 10035.
Children are taught to add multi-digit numbers from right to left, one digit at a time. Many find the "carry" operation, where a 1 is carried from one digit position to the next, to be a significant challenge. Your job is to count the number of carry operations for each of a set of addition problems so that educators may assess their difficulty.

Input

Each line of input contains two unsigned integers less than 10 digits. The last line of input contains "0 0".

Output

For each line of input except the last, compute the number of carry operations that result from adding the two numbers and print them in the format shown below.

Sample Input

```
123 456
555 555
123 594
0 0
```

Sample Output

```
No carry operation.
3 carry operations.
1 carry operation.
```

⋆

⋆ ⋆