

## PL1 - Travaux pratiques - Séance 1

### Premiers pas avec le langage C

Avant de commencer cette séance :

1. placez-vous dans votre répertoire **PL1** : `cd PL1`
2. récupérez les fichiers nécessaires à ce TP :  
`cp ~mounlaur/CCI_PL1/TP1.tar.gz .`
3. dé-compressez et dé-archiviez ce fichier :  
`gunzip TP1.tar.gz ; tar -xvf TP1.tar`
4. Placez-vous dans le répertoire **TP1** : `cd TP1`

### Exercice 1 - valeur absolue et signe du produit

Le fichier ***val\_absolue.c*** contient le source C d'un programme qui lit un entier (relatif) au clavier et affiche sa valeur absolue.

**Q1.** Essayez de compiler ce programme avec la commande suivante :

```
gcc -Wall -o val_absolue val_absolue.c
```

Le compilateur indique une erreur. Corrigez cette erreur (en ouvrant ***val\_absolue.c*** avec la commande ***nedit***) et compilez à nouveau pour produire l'exécutable ***val\_absolue***.

**Q2.** Exécutez le programme ***val\_absolue*** sur diverses entrées.

### Exercice 2 - signe du produit

On donne ci-dessous un algorithme qui lit au clavier deux entiers  $x$  et  $y$  et qui affiche à l'écran un message indiquant le signe du produit  $x \times y$  sans calculer ce produit.

**Signe du produit :**

```
x, y : entiers
lire(x) ; lire(y)
si ( ((x>0) et (y>0)) ou ((x<0) et (y<0)) )
alors   ecrire ("le produit de" & LeTexteE(x) & "et" LeTexteE(y) & "est strictement positif")
sinon   si ( (x=0) ou (y=0) )
        alors   ecrire ("le produit de" & LeTexteE(x) & "et" LeTexteE(y) & "est nul")
        sinon   ecrire ("le produit de" & LeTexteE(x) & "et" LeTexteE(y) & "est strictement negatif")
```

Le fichier ***signe\_produit.c*** contient le squelette d'un programme traduisant cet algorithme en langage C.

**Q1.** Compléter ce programme.

**Q2.** Le compiler avec la commande : `gcc -Wall -o signe_produit signe_produit.c`

**Q3.** Exécutez le programme *signe\_produit* sur diverses entrées de façon à traiter tous les cas possibles.

### Exercice 3 - classement de trois valeurs

On souhaite écrire un programme qui lit trois entiers au clavier et les affiche à l'écran en ordre croissant (du plus petit au plus grand).

**Q1.** Ecrire (sur papier, en notation algorithmique) un programme qui lit au clavier trois entiers  $x$ ,  $y$  et  $z$  et les affiche à l'écran en ordre croissant (on pourra s'aider de l'exercice E2.9 du cours d'Algorithmique).

**Attention**, il ne faut pas chercher à échanger les valeurs des variables  $x$ ,  $y$  et  $z$  (cette opération sera vue plus tard!).

**Q2.** En vous aidant de la fiche de traduction, écrivez dans un fichier *classer3V.c* une traduction en langage C de ce programme.

**Q3.** Compilez et testez ce programme. Combien de tests sont-il nécessaires au minimum pour vous assurer que toutes les instructions du programme ont été exécutées au moins une fois ?

**Q4.** Re-compilez votre programme avec l'option `-g` pour pouvoir l'exécuter sous **gdb** :

```
gcc -g -Wall -o classer3V classer3V.c
```

Combien d'instruction comporte votre programme en tout ? Combien d'instructions seront exécutées si vous lui donnez en entrée les valeurs 2, 9 et 9 ? Vérifiez-le avec **gdb** ....

### Exercice 4 - Puissance

On donne ci-dessous un algorithme qui lit au clavier deux entiers (positifs)  $x$  et  $y$  et affiche la valeur de  $x^y$  :

**Puissance :**

```
x, y : entiers
p : un entier  $\geq 0$  { valeur courante de  $x^y$  }
lire(x) ; lire(y)
p  $\leftarrow$  1
i parcourant [1..y]
    p  $\leftarrow$  p * x
ecrire(p)
```

**Q1.** En vous aidant de la fiche de traduction, écrivez dans un fichier *puissance.c* une traduction en langage C de cet algorithme.

**Q2.** Compilez et testez ce programme.

**Q3.** Modifiez-le pour prendre en compte des valeurs entières positives ou négatives pour  $x$  et  $y$ . Compilez et testez cette nouvelle version.

## Exercice 4 - Factorielle

**Q1.** Ecrivez dans un fichier *factorielle.c* un programme C qui lit un entier positif  $x$  au clavier et affiche la valeur de  $x!$ .

**Q2.** Compilez et testez votre programme. A partir de quelle valeur de  $x$  le résultat obtenu n'a t-il plus de sens ?

## Exercice 5 - Moyenne

Le fichier *moyenne.c* contient le source d'un programme C qui lit une séquence de 10 entiers au clavier et affiche la valeur moyenne de cette séquence.

**Q1.** Compilez ce programme et testez-le. Pour éviter de saisir à chaque exécution les 10 entiers attendus par le programme on pourra préparer des fichiers contenant différentes séquences de 10 entiers (un par ligne) et les fournir au programme par **redirection des entrées**.

**Q2.** Modifiez ce programme pour qu'il calcule maintenant la *moyenne olympique* des 10 entiers lus, c'est-à-dire la moyenne calculée en ne prenant pas en compte les valeurs minimales et maximales de la séquence d'entrée. Compilez et testez ce nouveau programme.