

Travaux Pratiques n°1

Cette séance de travaux pratiques contient un certain nombre d'exercice à faire sur papier.

Quelques commandes utiles en C

On présente deux commandes utiles `printf` et `scanf` qui permettent respectivement d'afficher un texte à l'écran et de lire une entrée au clavier. Ce sont donc deux commandes qui permettent au programme de communiquer. En informatique on appelle ces communications des entrées/sorties. Pour utiliser les commandes, on doit faire appel à une bibliothèque du langage C qui s'appelle `stdio` (standard input/output). Pour cela, on doit donc au début du programme écrire la commande :

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

Pour afficher du texte à l'écran, on utilise l'instruction `printf`

Exemples :

L'instruction : `printf("Bonjour \n");`
permet d'afficher : Bonjour

L'instruction : `printf ("Un entier :%d \n",3);`
permet d'afficher : Un entier : 3

Les caractères `\n` permettent un retour à la ligne après l'affichage.

L'instruction `scanf` permet de lire une entrée tapée sur le clavier pendant l'exécution du programme et de la stocker dans une variable préalablement déclarée.

Exemple :

```
int a ; //déclare une variable a de type entier
scanf("%d",&a) ; //lit l'entrée au clavier et la stocke dans a
```

Attention!!!!!!!

Ne pas oublier le “&” avant le nom de la variable.

Exercices d'algorithmique

Exercice 1

Ecrire l'algorithme permettant de résoudre l'équation du second degré $a \cdot x^2 + b \cdot x + c = 0$ à coefficients réels.

Exercice 2

Ecrire l'algorithme qui permet de calculer le nombre de parties à p éléments dans un ensemble à n éléments, appelé combinaisons. Lorsqu'on a n objets quelconques, la combinaison nous donne le nombre de possibilités d'en prendre p différents. On note cette combinaison C_n^p ou $\binom{n}{p}$, et elle est égale à $\frac{n!}{p!(n-p)!}$

1. Un exemple : on veut faire du sport 2 fois par semaine du lundi au vendredi. Ecrire tous les couples de jours possibles et vérifier que le nombre de possibilités correspond à ce que donne la formule.
2. Ecrire l'algorithme général permettant de calculer C_n^p
3. Le tester sur l'exemple donné ainsi que sur d'autres exemples.
4. Que se passe-t-il si $p > n$?

Exercice 3

Réaliser l'algorithme qui fait la chose suivante : Il lit un nombre réel, puis il affiche la somme de tous les nombres rentrés précédemment. L'algorithme s'arrête quand la somme est égale ou supérieure à 100.

Question bonus : au moment où l'algorithme s'arrête, il affiche également la moyenne arithmétique des nombres rentrés.

Langage C**Exercice 4**

Créez un répertoire **Informatique** dans votre répertoire personnel. Les exercices sur machine seront effectués dans ce répertoire.

1. Créer un fichier `hello.c` dans ce répertoire
2. Dans un éditeur de texte écrire :

```
#include <stdio.h>

int main()
{
    return(0);
}
```

3. Ecrire dans le `main` une petite instruction permettant d'afficher `Hello World !`
4. Ajouter des commentaires pertinents à ce programme
5. Utiliser la commande `indent` (dans le terminal) pour indenter automatiquement votre code

```
indent -kr hello.c -o helloKR.c
```

Regardez dans l'éditeur de texte la différence entre vos deux fichiers.

6. Compiler ce programme avec `gcc` (dans un terminal)
7. Lancer l'exécutable (dans un terminal)

```
./helloKR
```

Exercice 5

Écrire un programme dans lequel on définit une variable de type entier n que l'on initialise à 2 et qu'on l'affiche.

Exercice 6

Même chose que l'exercice précédent mais c'est à l'utilisateur du programme d'initialiser l'entier à la valeur de son choix.

Exercice 7

Écrire un programme qui demande à l'utilisateur d'initialiser deux entiers. Puis le programme stocke la somme de ces deux entiers dans une autre variable et enfin affiche le résultat.