

## Travaux pratiques n°2 (2h): Introduction au langage C (2)

### Objectifs

- Introduction au langage C
- Coder des algorithmes détaillés en langage C
- Découvrir les premières erreurs de compilation

### Documents nécessaires

Sujet du td2 d'algorithmique.

## Partie 1: Introduction pratique au langage C

### Objectifs

- Introduction au langage C (premier programme, compilation, exécution, variable, constante, affectation, entrées-sorties)

### Compléments sur les entrées-sorties en C

Nous avons vu dans le TP1 les premiers éléments pour faire des entrées-sorties en C. Vous trouverez ci-dessous quelques précisions sur **scanf** et **printf** qui vous seront nécessaires pour affiner vos programmes:

**printf(format,listevaleurs)** affiche la liste de valeurs (variables ou expressions) dans le format choisi. Le format est une chaîne de caractères entre guillemets (double quote "), comprenant un texte qui sera écrit tel quel, des spécifications de format (débutant par %) et des caractères spéciaux (\). A l'exécution, chaque spécification de format sera remplacée par une valeur de liste valeurs (dans l'ordre d'apparition des formats).

Une spécification de format est de la forme :

`%[largeur][.précision]type` (entre [] facultatifs)

La **largeur** est le nombre minimal de caractères à écrire (des blancs sont rajoutés si nécessaire). Si le texte à écrire est plus long, il est néanmoins écrit en totalité. Attention dans le cas d'un réel le point (.) est compté.

La **précision** définit, pour les réels, le nombre de chiffres après la virgule (doit être inférieur à la largeur). Dans la cas d'entiers, indique le nombre minimal de chiffres désiré (ajout de 0 sinon).

Le **type** est : c (char), s (chaîne de caractères, jusqu'au \0), d (int), u (entier non signé), x ou X (entier affiché en hexadécimal), o (entier affiché en octal), f (réel en virgule fixe), e ou E (réel en notation exponentielle), g ou G (réel en f si possible, e sinon), p (pointeur), % (pour afficher le signe %).

Les caractères spéciaux utilisables dans le format sont \t (tabulation), \n (retour à la ligne), \\ (signe \).

exemple:

```
printf("moyenne calculée :%5.2f\n",moyenne);
```

Affiche la moyenne sur 5 chiffres minimums avec 2 chiffres après la virgule (ex: 12.45 ou 02.50)

**scanf(format,listeadresse)** lecture au clavier de valeurs, dans le format spécifié. Les arguments sont les adresses des variables résultats. **scanf** retourne le nombre de valeurs effectivement lues et mémorisées.

## Le casting: introduction

Il est possible que lors d'un calcul les opérandes n'ont pas le même type ou que le type du résultat de l'opération que vous souhaitez obtenir n'est pas celui des opérandes. Il faut donc expliciter le type désiré par la syntaxe suivante: *(type) expression*.

Ainsi, par exemple supposons la division de 1 par 2. Par défaut, le langage C effectuera une division entière arrondie à l'entier inférieur. Ainsi 1/2 donne comme résultat 0. Si vous voulez obtenir le résultat sous forme d'un float, vous devez écrire:

```
(float) 1/2;
```

**exemple:**

```
#include <stdio.h>
int main(void)
{
    float un;
    int n=2;

    un = 1;
    un= un + (float) 1/(n*n);
```

---

```
printf("\n un = %f", un);
return(1);
}
```

**va afficher:**

un = 1.25

**Alors que**

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

```
int main(void)
```

```
{
```

```
float un;
```

```
int n=2;
```

```
un = 1;
```

```
un= un + 1/(n*n);
```

```
printf("\n un = %f", un);
```

```
return(1);
```

```
}
```

**va afficher:**

un = 1

## Partie 2: Traduire des corps d'algorithmes détaillés en langage C

### Objectifs

- Créer ses premiers programmes.
- Découvrir les premières erreurs de compilation

### Exercice 1

Ecrire, compiler et exécuter un fichier *peinture.c* traduisant en langage C l'algorithme de l'exercice 2 du td2

### Exercice 2

Donner le programme en C qui fait dessiner la figure suivante à la tortue en demandant la longueur d'une branche (coté du carré) à l'utilisateur. A l'initialisation on supposera que la tortue est placée quelque part sur l'écran et regardant vers le haut.



