3 Les variables (2nde partie)

Durant tout ce chapitre, nous aurons pour but simple, mais complet et complexe dans sa programmation, de faire une malheureuse calculatrice.

3.1 Exercice de mise en bouche

Ecrire un programme qui :
- Ecrit "Calculatrice :" **et** saute 2 lignes.
- Ecrit "Valeur de a :" **et** saute 1 ligne.
- Attend l'appui d'une touche.
- Ecrit "Valeur de b :" **et** saute 1 ligne.
- Attend l'appui d'une touche
- Ecrit "Valeur de a + b :"

Pas à pas, nous allons maintenant réaliser notre petit programme de calculatrice.

3.2 Déclaration des variables

Complétez le programme en :
- Déclarant 2 variables a et b de type int (entier).
- Assignant à ces deux variables les valeurs de 36 et 54.
- Faisant afficher le résultat de la somme de a + b (et non pas afficher 90 !).

Pour faire afficher le résultat il est possible d'utiliser la fonction printf directement ou indirectement en utilisant une troisième variable. Pour plus de facilité, nous allons utiliser un affichage direct. Celui-ci s'effectue de la façon suivante :

N.B. printf ("Valeur de a + b : %d",a+b);
Le %d est remplacé par la valeur de a+b

3.3 Saisie des variables

Si une calculatrice se limitait à exécuter la somme de deux nombres fixes, le boulier serait encore de mise …

Pour saisir une variable, il est nécessaire d'utiliser la fonction scanf. La fonction scanf s'utilise de la façon suivante :

Saisie de la valeur a : **scanf ("%d", &a);**

Comme pour printf, on reconnaît le %d pour la saisie d'un nombre décimal. Le & devant le a signifie que l'on va écrire dans la variable a.

En fait &a signifie l'adresse mémoire de a. Scanf va donc écrire dans l'emplacement mémoire (la petite boîte contenant la variable) de a. Si l'on oublie le &, on écrira chez quelqu'un d'autre, à une autre adresse. Et là ça peut faire mal, très mal …
Mais ceci est une autre histoire …

Nous allons maintenant saisir les variables a et b. Pour exemple, je mets ici le code pour la saisie de a, la saisie de b reste à faire par vos soins à titre d'exercice.

```
/* Saisie de la valeur de a */
printf ("Valeur de a :\n");
scanf ("%d",&a);
```

Il est conseillé d'initialiser les variables avant de les utiliser, de ce fait, au début du programme, après leur déclaration, assignez à a et à b la valeur de 0.

Exemple :

```
a = 0;
```

Si tout c'est bien passé, vous avez maintenant entre les mains une super calculatrice

N.B. *Appuyez sur la touche Enter après avoir taper votre valeur.*

3.4 Déclaration avec initialisation d'une variable

Pour aller plus rapidement, il est possible d'initialiser une variable dans le même temps que sa déclaration. Pour cela, rien de plus simple, ajouter à la fin de la déclaration = suivi de la valeur d'initialisation.

Exemple :

```
int i = 10;
```

3.5 Evolution du logiciel : exercice

Aïe Aïe Aïe Dur Dur d'être informaticien. J'ai envie de faire une super calculatrice et je me dis qu'en fait la simplicité qui a eu lieu tout à l'heure n'est pas forcément adapté à

- Assigner une troisième valeur de type int (penser à l'initialiser à 0) que l'on nommera bêtement s comme somme.
- Une fois les valeurs de a et b saisies, initialisez s avec la valeur de a + b. Comme en mathématiques élémentaires.
- Affichez la valeur de s. On devrait avoir les même résultats qu'auparavant, off course.

3.6 Exercices d'application

Réalisez 2 petits programmes qui font :

- la soustraction de 2 nombres
- la multiplication de 2 nombres

3.7 Un nouveau type: les nombres à virgule ou flottants (float)

Le type **float** permet de déclarer un nombre à virgule. Transformez les 3 précédents programmes en utilisant le type float au lieu du type int et %f pour saisir les valeurs de a et b, et pour afficher le résultat.

Petite aide :

```
float    a;
printf ("Saisie de a :");
scanf ("%f",&a);
printf ("\na vaut : %f",a);
getch ();
```

Ce petit morceau de programme saisit la valeur de a et la fait afficher.

Pour un affichage plus agréable il est possible de fixer le nombre de chiffres après la virgule de la façon suivante **%.[nombre de chiffres après la virgule]f**.

Exemple :

```
printf ("%.2f",a);
```

3.8 Ouah, les 4 opérations !!!

Créer un quatrième programme qui réalise la division de deux nombres.
Tester avec une division par 0 !!!
La solution à ce petit problème sera vu dans le cours des conditions (if).

3.9 Exercices à réaliser

Compléter les 4 programmes afin de réaliser les opérations suivantes :

$$\begin{cases} s = a + b + c \\ s = a - b - c \\ s = a / b / c \\ s = a * b * c \end{cases}$$

où a,b,c sont des nombres à virgules (float).

La fonction **abs** permet d'obtenir la valeur absolue d'un nombre entier. Utiliser cette fonction pour calculer la valeur absolue de (a-b).
Exemple d'utilisation de abs :

```
S = abs (a-b);
```

La fonction **ceil** permet d'obtenir l'arrondi entier supérieur d'un nombre réel. Utiliser cette fonction pour calculer l'arrondi supérieur de (a / b).
Exemple d'utilisation de ceil :

```
S = ceil (a/b);
```

Corrigés des exercices du chapitre 3

! *Soustraction de 2 nombres*

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

int main ()
{
   int a,b;               /* Déclaration des variables */
   int d;                 /* Différence entre les 2 nombres */

   a=0;                   /* Initialisation des variables */
   b=0;

   clrscr ();
   printf("Calculatrice :\n\n");
   printf("Valeur de a : ");
   scanf("%d",&a);
   printf("\n");
   printf("Valeur de b : ");
   scanf("%d",&b);

   d=a-b;
   printf("Valeur de a-b : %d\n",d);  /* Affichage de la différence */

   getch ();
   return 0;
}
```

! *Multiplication de 2 nombres*

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

int main ()
{
   int a,b;               /* Déclaration des variables */
   int m;                 /* Résultat de la multiplication */
   a=0;                   /* Initialisation des variables */
   b=0;

   clrscr ();
   printf("Calculatrice :\n\n");
   printf("Valeur de a : ");
   scanf("%d",&a);
   printf("\n");
   printf("Valeur de b : ");
   scanf("%d",&b);

   m = a*b;
   printf("Valeur de a*b : %d\n", m);
   /* Affichage de la multiplication */

   getch ();
   return 0;
}
```

! ***Transformez les 3 précédents programmes avec le type float***

Aucune difficulté dans cet exercice, je ne mettrai ici la correction que de la multiplication, les autres opérations se réalisent sur le même schéma.

```
int main ()
{
   float a,b;              /* Déclaration des variables */
   float m;                /* Résultat de la multiplication */

   a=0;                    /* Initialisation des variables */
   b=0;

   clrscr ();
   printf("Calculatrice :\n\n");
   printf("Valeur de a : ");
   scanf("%f",&a);
   printf("\n");
   printf("Valeur de b : ");
   scanf("%f",&b);

   m = a*b;
   printf("Valeur de a*b : %f\n", m);
   /* Affichage de la multiplication */

   getch ();
   return 0;
}
```

Pour l'addition :

```
   m = a + b;
   printf ("Valeur de a+b : %f", m);
```

Pour la soustraction :

```
   m = a - b;
   printf ("Valeur de a-b : %f", m);
```

! ***Calculer la somme a + b + c***

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

int main ()
{
   float a,b,c,s;              /* Déclaration des variables */

   a=0;                        /* Initialisation des variables */
   b=0;
   c=0;
   s=0;

   clrscr ();
   printf("Saisie de a : ");
   scanf("%f",&a);
   printf("Saisie de b : ");      /* Saisie variables flottantes */
   scanf("%f",&b);
   printf("Saisie de c : ");
   scanf("%f",&c);
```

```
   s = a+b+c;                          /* Calcul de la somme */
   printf("Valeur de s : %.2f\n",s);   /* Affichage de la somme */

   getch ();
   return 0;
}
```

! ***Calculer la différence a - b - c***

```
   d = a-b-c;                          /* Calcul de la différence */
   printf("Valeur de d : %.2f\n",d);   /* Affichage de la différence */
```

! ***Calculer la multiplication a * b * c***

```
   m = a*b*c;                          /* Calcul de la multiplication */
   printf("Valeur de m : %.2f\n",m);   /* Affiche la multiplication */
```

! ***Calculer la division a / b / c***

```
   d = a/b/c;                          /* Calcul de la division */
   printf("Valeur de d : %.2f\n",d);   /* Affichage de la division */
```

! ***Calculer la valeur absolue de a - b***

```
   r=abs(a-b);                         /* Calcul de la valeur absolue */
   printf("Valeur de r : %f\n",r);     /* Affiche la valeur absolue */
```

! ***Calculer l'arrondi de a + b***

```
   c=ceil(a/b);                        /* Calcul de l'arrondi */
   printf("Valeur de c : %f\n",c);     /* Affichage de l'arrondi */
```

N Il aurait été possible d'utiliser **%d** du fait que l'arrondi est un nombre entier !