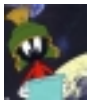
2 Les variables (1ère partie)

2.1 Printf : fonction indispensable pour afficher une variable

Exemple :

```
#include <stdio.h>

int main ()
{
        printf ("Coucou c'est moi");   /* Affiche Coucou c'est moi à l'écran */
        getch ();                      /* Attendre l'appui d'une touche */
}
```



On pourrait dire que la fonction **printf** est la même que l'instruction **puts** vu précédemment mais il n'en est rien … Celle-ci est beaucoup, beaucoup, beaucoup plus puissante.

Syntaxe:

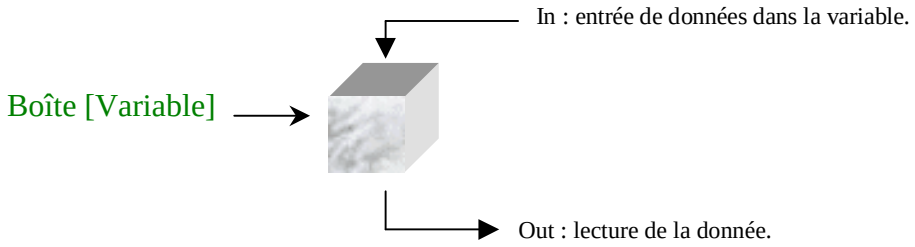
La syntaxe de printf est très complexe et pourrait à elle seule faire l'objet d'un cours, nous en verrons donc des applications au fur et à mesure des besoins.

2.2 Variable

Comme son nom l'indique une variable est quelque chose qui varie.
C'est vrai mais ce n'est pas suffisant.

Une variable peut être considérée comme une boîte dans laquelle on met des données que l'on peut lire ou écrire.

Variable : Lire ou Ecrire



La manière la plus facile de lire le contenu d'une variable est la fonction **printf** que l'on a aperçu précédemment.

La manière la plus simple de donner une valeur à une variable est l'opérateur mathématique =. Ecrire dans une variable ayant déjà une valeur revient à la modifier.

Une variable ne peut contenir qu'une seule chose à la fois. Si vous mettez une seconde donnée dans la variable, la précédente est effacée.

2.3 Déclaration d'une variable

La déclaration d'une variable se fait simplement en écrivant :

<son type> <son nom>;

Exemples de type de variables :

char : caractère
int: entier

2.4 Application : exemple

```
#include <stdio.h>

int main ()
{
 int      i;      /* i : variable de type entier */
 char     car;    /* car: variable de type caractère */

 i = 65;          /* i vaut 65 */
 car = 'E';       /* car vaut E */

 clrscr ();       /* Efface l'écran */

 printf ("i vaut %d.\n", i);       /* Affiche la valeur de i */
 printf ("car vaut %c.\n",car);    /* Affiche la valeur de car */

 getch ();                         /* Attendre l'appui d'une touche */

 return (0);
}
```

Déclarations des variables

Ecriture dans les variables

Affichage des variables

Explications :

On met dans la variable i la valeur 65.
On met dans la variable car la valeur de E.

Note : En informatique, tout n'est que nombre, je dis donc la valeur de E et non E car c'est le code Ascii de E qui est sauvegardé dans cette variable. Nous reviendrons là dessus un peu plus tard.

- printf ("i vaut **%d**.\n", i);
  %d signifie que l'on attend une **valeur décimale**, le programme va donc remplacer le %d par la valeur de i.

- printf ("car vaut **%c**.\n", car);
  %c signifie que l'on attend une **valeur de type caractère**, le programme va donc remplacer le %c par la valeur de car.

- **\n** permet de réaliser un retour chariot c'est à dire un retour à la ligne.

2.5 Utilisation multiple du %

Le code "%*x*" signifie que le compilateur C doit **remplacer** ce code par la valeur correspondante (qui lui est fourni dans la suite de l'instruction) en la **transformant** dans le type *x*. Cette transformation est appelé un **cast**.

Exemple :

```
int i;
i =65;
printf ("Le caractère %d est %c",i,i);
```

nous donnera l'affichage suivant :

Le caractère 65 est A.

Le %d est remplacé par la valeur numérique de i c'est à dire 65.
Le %c est remplacé par la valeur alphanumérique (ASCII) de i c'est à dire A.
cf. Table Ascii en Annexe.

2.6 Exercices d'applications directes

En utilisant ce qui a été fait précédemment, faîtes afficher les valeurs 70, 82, 185 et 30.

En utilisant ce qui a été fait précédemment, faîtes afficher, les caractères c, o, u, C, O, U.

2.7 Réutilisation d'une variable

On peut réutiliser une variable autant de fois que l'on veut, la précédente valeur étant effacée.

i = 3; i = 5; i = 7; donnera au final pour valeur de i la valeur de 7.

car = 'E'; car = 'G'; car = 'h'; donnera au final pour valeur de car la valeur de 'h'.

2.8 Caractères spéciaux

Certains caractères utilisés par la fonction printf ("%" par exemple) ou même tout simplement pour déclarer une variable (' pour les caractères par exemple) oblige à utiliser le caractère de suffixe \ pour pouvoir être affiché.

Exemple :

- Pour afficher un % avec printf j'écrirai :

```
printf "La réduction était de 20 \%"
```

- Pour déclarer un caractère avec la valeur ' (prononcée cote en informatique et non pas apostrophe (français)), on écrira :

```
char car;
car = '\'';
```

2.9 Exercices à réaliser

Faire un programme qui réalise l'affichage suivant en utilisant les variables
Aide : Sans retour chariot, on affiche à la suite

Coucou
17

De la même façon, réaliser un programme qui réalise l'affichage suivant :

C
O
U

Cou
1
2
3
456
C'est rigolo …

Rappel : pour mettre une variable **c** égale à ' on écrit **c =** '\''

Ecrire un programme qui écrit
« Hamlet says "To be or not to be, that is the question." »
avec les " bien sûr.

Rappel : Pour pouvoir afficher un caractère de syntaxe C, par exemple ", on utilise le caractère \ comme préfixe à ce caractère. Pour obtenir un ", on utilise donc \ ".

Corrigés des exercices du chapitre 2

! ***Faites afficher, en utilisant ce qui a été fait précédemment, les valeurs 70, 82, 185 et 30.***

```
#include <stdio.h>

int main ()
{
   int i,a,b,c;
   i=70;
   a=82;
   b=185;
   c=30;
   clrscr ();
   printf ("i vaut %d.\n",i);
   printf ("a vaut %d.\n",a);
   printf ("b vaut %d.\n",b);
   printf ("c vaut %d.\n",c);
   getch ();

   return 0;
}
```

! ***Faites afficher, en utilisant ce qui a été fait précédemment, les caractères c, o, u, C, O, U.***

```
#include <stdio.h>

int main ()
{
   char a,b,c,d,e,f;
   a='c';
   b='o';
   c='u';
   d='C';
   e='O';
   f='U';
   clrscr ();
   printf ("a vaut %c.\n",a);
   printf ("b vaut %c.\n",b);
   printf ("c vaut %c.\n",c);
   printf ("d vaut %c.\n",d);
   printf ("e vaut %c.\n",e);
   printf ("f vaut %c.\n",f);

   printf ("a, b, c, d, e, f
   valent : %c, %c, %c, %c, %c,
   %c.\n",a,b,c,d,e,f);

   getch ();
   return (0);
}
```

! ***Faire un programme qui réalise l'affichage …***

Coucou
17

```
#include <stdio.h>

int main ()
{
      char  car = 'C';
      int   nbre = 17;

      printf ("%c",car);
      car = 'o';
      printf ("%c",car);
      car = 'u';
      printf ("%c",car);
      car = 'c';
      printf ("%c",car);
      car = 'o';
      printf ("%c",car);
      car = 'u';
      printf ("%c",car);

      printf ("\n%d",nbre);

      /* Attente */
      getch ();

      return (0);
}
```

! ***Faire un programme qui réalise l'affichage …***

C
O
U
…

```
#include <stdio.h>

int main ()
{
      char  car = 'C';
      int   nbre = 1;

      clrscr (); /* Efface l'écran
*/
      car = 'C';
      printf ("\n%c",car);
      car = 'O';
      printf ("\n%c",car);
      car = 'U';
      printf ("\n%c",car);
      car = 'C';
      printf ("\n%c",car);
      car = 'o';
      printf ("%c",car);
      car = 'u';
      printf ("%c",car);

      printf ("\n%d",nbre);
      nbre = 2;
      printf ("\n%d",nbre);
      nbre = 3;
      printf ("\n%d",nbre);
      nbre = 456;
      printf ("\n%d",nbre);
}
```

```
      car = 'C';
      printf ("\n%c",car);
      car = '\'';
      printf ("%c",car);
      car = 'e';
      printf ("%c",car);
      car = 's';
      printf ("%c",car);
      car = 't';
      printf ("%c",car);
      car = ' ';
      printf ("%c",car);
      car = 'r';
      printf ("%c",car);
      car = 'i';
      printf ("%c",car);
      car = 'g';
      printf ("%c",car);
      car = 'o';
      printf ("%c",car);
      car = 'l';
      printf ("%c",car);
      car = 'o';
      printf ("%c",car);
      /* Attente */
      getch ();

      return (0);
```

! ***Faire un programme qui écrit "Hamlet says … "***

```
#include <stdio.h>

int main ()
{
      clrscr ();  /* Efface l'écran */

      printf ("Hamlet says : \"To be or no to be, that is the
      question.\"");

      /* Attente */
      getch ();

      return (0);
}
```