

CHAPITRE 2

SAISIE DE NOMBRES ET DE CARACTERES AU CLAVIER

LA FONCTION GETCH

La fonction `getch`, appartenant à la bibliothèque `conio.h` permet la saisie clavier d' un caractère alphanumérique, **sans écho écran**. La saisie s'arrête dès que le caractère a été frappé.

La fonction `getch` n'est pas définie dans la norme ANSI mais elle peut exister dans la bibliothèque d'autres compilateurs.

On peut utiliser `getch` de deux façons:

- sans retour de variable au programme:

Exemple:

```
printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
getch();
```

- avec retour de variable au programme:

Exemple:

```
char alpha;
printf("ENTRER UN CARACTERE (ATTENTION PAS DE RETURN) ");
alpha = getch();
printf("\nVOICI CE CARACTERE: %c",alpha);
```

Les parenthèses vides de `getch()` signifient qu'aucun paramètre n'est passé à cette fonction par le programme.

LA FONCTION SCANF

La fonction `scanf`, appartenant à la bibliothèque `stdio.h`, permet la saisie clavier de n'importe quel type de variable.

Les variables à saisir sont formatées, le nom de la variable est précédé du symbole `&` désignant l'adresse de la variable (On reverra ce symbole dans le chapitre sur les pointeurs). La saisie s'arrête avec "RETURN" (c'est à dire LF), les éléments saisis s'affichent à l'écran (**saisie avec écho écran**).

Tous les éléments saisis après un **caractère d'espace** (espace, tabulation) sont ignorés.

Exemples:

```
char alpha;
int i;
float r;
scanf("%c",&alpha);      /* saisie d'un caractère */
scanf("%d",&i);          /* saisie d'un nombre entier en décimal */
scanf("%x",&i);          /* saisie d'un nombre entier en hexadécimal*/
scanf("%f",&r);          /* saisie d'un nombre réel */
```

Remarque: Si l'utilisateur ne respecte pas le format indiqué dans scanf, la saisie est ignorée. Aucune erreur n'est générée.

Exemple:

```
char alpha;
scanf("%d",&alpha);
```

Si l'utilisateur saisie 97 tout va bien, alpha devient le caractère dont le code ASCII vaut 97. Si l'utilisateur saisie a, sa saisie est ignorée.

Exercice II 1:

Saisir un caractère au clavier, afficher son code ASCII à l'écran. Soigner l'affichage.

Exercice II 2: Saisir un nombre entier en décimal au clavier, l'afficher en hexadécimal à l'écran. Soigner l'affichage.

Exercice II 3:

Que va-t-il se passer lors de l'exécution du programme suivant, si l'utilisateur saisit 67?

```
#include <stdio.h>
#include <conio.h>
void main()
{
char c;
printf("ENTRER UN CARACTERE: ");
scanf("%c",&c);
printf("VOICI SON CODE ASCII: %d\n",c);
printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
getch();
}
```

NOTION DE FLUX D'ENTREE

Lorsque l'on saisit au clavier une suite de caractères terminés par "RETURN" ces caractères sont rangés dans un tampon (ou buffer) de type FIFO (First In/First Out), le dernier caractère rangé dans le tampon est LF (code ASCII 0x0A).

Cette suite de caractères est appelée **flux d'entrée**.

La taille du tampon dépend de la machine et du compilateur utilisés. Sur un PC et en TURBOC, la taille du tampon est de 127 caractères.

Une compilation du programme vide le tampon.

**LA FONCTION SCANF
DEUXIEME APPROCHE**

La fonction scanf ne se comporte pas tout à fait comme décrit plus haut. Si le tampon est vide, tout se passe comme précédemment décrit.

Au contraire, si le tampon n'est pas vide, la fonction scanf en teste le premier élément, s'il correspond au format de la variable invoquée, le tampon perd cet élément et la variable en prend la valeur.

Tout caractère ou nombre saisi au clavier et non pris en compte par la fonction scanf est rangé dans le tampon.

Exemple et Exercice II-4:

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char c1,c2;
  printf("ENTRER UN CARACTERE: ");
  scanf("%c",&c1);
  printf("VOICI SON CODE ASCII EN HEXADECIMAL: %x\n",c1);
  printf("ENTRER UN AUTRE CARACTERE: ");
  scanf("%c",&c2);
  printf("VOICI SON CODE ASCII EN HEXADECIMAL: %x\n",c2);
  printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
  getch();
}
```

Si l'utilisateur saisit K pour c1, le programme donnera l'écran d'exécution suivant:

```
ENTRER UN CARACTERE: K
VOICI SON CODE ASCII EN HEXADECIMAL: 4b
ENTRER UN AUTRE CARACTERE: VOICI SON CODE ASCII EN HEXADECIMAL: a
```

Lors de la saisie de K, le caractere LF est rangé dans le tampon. Lors du deuxième appel à scanf, le tampon n'est pas vide, l'utilisateur ne peut effectuer sa saisie clavier, le code ascii de LF est affiché à l'écran.

A l'issue de l'exécution, le tampon est vide.

Exercice II 5: Le programme suivant s'exécute-t-il "correctement" ? Que contient le tampon à l'issue de l'exécution ?

```
#include <stdio.h>
#include <conio.h>
void main()
{
char c;
int i;
printf("ENTRER UN CARACTERE: ");
scanf("%c",&c);
printf("VOICI SON CODE ASCII EN HEXADECIMAL: %x\n",c);
printf("ENTRER UN NOMBRE: ");
scanf("%d",&i);
printf("VOICI CE NOMBRE EN HEXADECIMAL: %x\n",i);
printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
getch();
}
```

Exercice II 6: Le programme suivant s'exécute-t-il "correctement" ? Que contient le tampon à l'issue de l'exécution ?

```
#include <stdio.h>
#include <conio.h>
void main()
{
char c;
int i;
printf("ENTRER UN NOMBRE: ");
scanf("%d",&i);
printf("VOICI CE NOMBRE EN HEXADECIMAL: %x\n",i);
printf("ENTRER UN CARACTERE: ");
scanf("%c",&c);
```

```
printf("VOICI SON CODE ASCII EN HEXADECIMAL: %x\n",c);
printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
getch();
}
```

Exercice II_7: Dans l'exercice II_4 que se passe-t-il si, lors de la première saisie, l'utilisateur tape 67 ?

Remarque: En TURBOC la fonction **flushall()** permet de vider le tampon d'entrée. En l'invoquant après un appel à scanf, on se débarrasse des problèmes de flux d'entrée.

LA FONCTION GETCHAR

La fonction getchar permet la saisie d'un caractère (char). Elle appartient à la bibliothèque stdio.h. Les 2 écritures suivantes sont équivalentes:

char c;	char c;
printf("ENTRER UN CARACTERE: ");	printf("ENTRER UN CARACTERE: ");
scanf("%c",&c);	c = getchar();

Non formatée, la fonction getchar est moins gourmande en place mémoire que scanf. Il vaut mieux l'utiliser quand cela est possible; getchar utilise le flux d'entrée exactement comme scanf.

CORRIGE DES EXERCICES

Exercice II_1:

```
#include <stdio.h>
#include <conio.h>
void main()
{
char c;
printf("ENTRER UN CARACTERE: ");
scanf("%c",&c);
printf("VOICI SON CODE ASCII EN DECIMAL: %d\n",c);
puts("Pour continuer frapper une touche...");
getch();
}
```

Exercice II 2:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int nombre;
    printf("ENTRER UN NOMBRE ENTIER: ");
    scanf("%d",&nombre);
    printf("VOICI CE NOMBRE EN HEXADECIMAL: %x\n",nombre);
    printf("POUR CONTINUER FRAPPER UNE TOUCHE ");
    getch();
}
```

Exercice II 3: Seul le **caractère 6** est pris en compte. L'affichage suivant la saisie donnera 54, c'est à dire le code ASCII de 6.

Exercice II 5: Oui car lors du deuxième appel à scanf, le programme attend un entier (int), alors que le tampon ne contient qu'un caractère (char).
A l'issue de l'exécution le tampon contient les deux caractères LF.

Exercice II 6: Non car à l'issue de la première saisie, le tampon contient le caractère LF qui sera lu lors du deuxième appel à scanf. Après exécution du programme, le tampon est vide.

Exercice II 7: L'affichage de c1 en hexadécimal donne 36 c'est à dire le code ASCII de 6, l'utilisateur ne peut saisir c2, l'affichage de c2 en hexadécimal donne 37 c'est à dire le code ASCII de 7.