IN200 : Entrées-Sorties

Sandrine Vial
sandrine.vial@uvsq.fr

Mars 2016

Contrôle Continu

- Mercredi 23 Mars Amphi D.
- Une feuille A4 recto-verso autorisée.

Entrées-Sorties

- Instructions pour :
  - **Entrer** des données à partir du clavier (de la souris, ...)
  - **Afficher** des données sur l'écran (ou autres périphériques de sortie)
- Pour quoi faire ?
  - Donner au programme les données sur lesquelles il va travailler. Par exemple, donner des valeurs à des variables.
  - Afficher les résultats d'un calcul ou la valeur d'une variable.

Entrées-Sorties

- scanf : donner des valeurs entrées au clavier à des variables passées en argument de la fonction.
- printf : imprimer à l'écran des valeurs données en argument.

La fonction `printf`

- On peut mélanger textes fixes et valeurs de variables :

```
printf("La valeur de la variable i est %d \n",i);
printf("Addition : %d + %d = %d",i,j,i+j);
```

- La chaine de caractères du premier argument sert à :
  - Imprimer du texte fixe
  - Indiquer le format des variables passées dans les arguments suivants.

La fonction `printf`

`printf(format,liste d'expressions)`

- Le premier argument est une chaine de caractères contenant textes fixes et formats
- Les arguments suivants sont des expressions
- La substitution se fait par position

La fonction `printf`

Les formats précédés de % :

- c : `char` un caractère
- d : `int` un entier
- f : `float` un réel en écriture décimale
- e : `float` un réel en notation exp.

La fonction `printf`

Le comportement est le suivant :

- Les expressions sont évaluées
- Les codes de formats sont remplacés par les valeurs des expressions correspondantes
- La chaine est ensuite affichée.

Exemple

```
#include <stdio.h>

int main()
{
int i,j;
float f;

i = 10;
j = 1456;
f = 34.5;

printf("A:\n%d\n%d\n%f\n",i,j,f);
printf("B:\n%4d\n%4d\n%4.2f\n",i,j,f);
printf("C:\n%6d\n%6d\n%9.2f\n",i,j,f);
printf("D:\n%04d\n%04d\n%f\n",i,j,f);
printf("E:\n%x\n%x\n%f\n",i,j,f);
printf("F:\n%o\n%o\n%f\n",i,j,f);
printf("G:\n%f\n%f\n%f\n",i,j,f);
return 0;
}
```

Entrées : la fonction `scanf`

- `scanf(format,liste d'adresses)`
- Algorithme appliqué à la lecture des formats :
  - Chaque directive du format est exécutée. Si une échoue on retourne à la fonction appelante.
  - Renvoie le nombre d'éléments correctement mis en correspondance et `EOF` en cas de fin de flux ou d'erreur.

Des directives

- des caractères blancs
- des caractères ordinaires
- %d : entier
- %f : réel
- %c : caractère
- %s : chaine de caractères

Directives

- Directive caractère blanc : consomme tous les caractères (même un nombre vide)
- Directive caractère quelconque : doit être identique au flot d'entrée.
- Directive % : autres %c consomment les blancs avant les autres.
- Directive %s : s'arrête au premier caractère blanc rencontré.

Exemple

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
int n,nb;
char c,d;

nb = scanf("%d%c%c",&n,&c,&d);
printf("nbe de valeurs lues : %d \n",nb);
printf("Contenu de n : <%d>\n",n);
printf("Contenu de c : <%c>\n",c);
printf("Contenu de d : <%c>\n",d);
nb = scanf(" %c",&c);
printf("nbe de valeurs lues : %d \n",nb);
printf("Contenu de c : <%c>\n",c);
printf("FIN\n");
return 0;
}
```