

Initiation au langage C

2- E/S et structures de contrôle (suite), fonctions



Objectifs : Être capable de développer des applications basiques de type console en C. Être capable d'utiliser les fonctions.

Avertissement

Tout au long de ce TP, vous répondrez aux questions posées sur votre compte-rendu. Ce compte-rendu devra inclure les programmes réalisés. Ces programmes devront être commentés !

Affectation et tests

L'instruction d'affectation est plus complexe qu'une simple égalité. En C, quand on écrit $a = b$, il faut lire cela comme la variable a reçoit le contenu de la variable b . Dans d'autres langages informatique, le symbole est parfois différent ainsi, en pseudo code on écrit $a \leftarrow b$, en Pascal l'affectation s'écrit $a := b$.

Enfin, attention aussi à ne pas confondre l'affectation avec un test d'égalité.

L'instruction qui permet de tester si deux variables sont égales s'écrit en C : `if (a == b)`



Travaux dirigés

Q.1) Déterminer les valeurs de a et b après les instructions suivantes. Conclure !

Programme 1 :	<code>a = 3 ;</code>	Programme 2 :	<code>a = 3 ;</code>
	<code>b = 1 ;</code>		<code>b = 1 ;</code>
	<code>a = b ;</code>		<code>b = a ;</code>
	<code>b = a ;</code>		<code>a = b ;</code>

Q.2) Écrire un algorithme appelé Permut qui réalise les opérations suivantes :

Saisie d'une variable entière a , puis d'une variable entière b .

Permutation de a et b puis affichage des nouvelles valeurs de a et b après permutation.



Manipulation n°1

Q.1) Écrire et tester le programme correspondant à l'algorithme Permut écrit précédemment en travaux dirigés.

Q.2) Créer un projet appelé Compar qui réalise les opérations suivantes :

Saisie d'une variable a , puis d'une variable b .

Comparaison de a et b et affichage du résultat.

La variable a est (supérieure, égale ou inférieure) à la variable b

Affichage des données



Manipulation n°2

Q.1) Écrire un projet qui permet de saisir une variable de type char (%c) puis l'afficher sous la forme d'un caractère (%c), d'un entier (%d), en hexadécimal (%x) et d'un réel (%f).

Q.2) Essayer votre programme puis conclure.

Q.3) Mêmes questions mais on saisit maintenant un entier (%d).

Q.4) Mêmes questions mais on saisit maintenant un réel (%f).

Q.5) Conclure

Les mystères de la division !



Manipulation n°3

Q.1) Écrire un projet qui demande de rentrer deux entiers puis réalise la division de ces deux entiers et affiche le résultat. Attention, la division de deux entiers peut donner un résultat réel ($2/3 = 0,6666\dots$).

Il y a plusieurs solutions mais elles ne sont pas « évidentes » ! Faites-vous aider par l'ami Google !

Boucles

Table de multiplication



Manipulation n°4

Q.1) Écrire un programme qui affiche la table de multiplication (de 0 à 10) d'un entier rentré au clavier. On utilisera la structure for.

Bonus : si vous êtes en avance, faire quelque chose de joli avec les nombres bien alignés en utilisant les formats.

Calcul de π

La relation suivante dite formule de Machin permet de calculer le nombre π .

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} \dots \right)$$

Le programme que vous devrez écrire doit calculer le nombre π en utilisant cette formule.

Algorithme

On propose un algorithme mettant en œuvre cette formule.

```
-----  
Algorithme : Calcul de pi  
-----  
Variables  
entiers : i, signe  
réels : somme, precision, pi, terme  
  
    DEBUT  
        Afficher "Calcul de Pi par la formule de machin"  
        Afficher "Précision désirée : "  
        Saisir precision  
        i ← 1  
        Somme ← 0  
        Signe ← 1  
        Répéter  
            terme ← 1/i  
            somme ← somme + (signe * terme)  
            signe ← - signe  
            pi ← 4* somme  
            Afficher " Pi calculé = ";pi ; dernier terme = ";terme)  
            i ← i+2  
        Jusqu_a (terme > 0,5*precision)  
    FIN
```



Travaux dirigés

Décrire ce qui se passe lorsque cet algorithme est exécuté et détailler les valeurs des variables pour les 3 premières itérations.

Quel est le rôle de l'instruction `signe ← - signe` ?

Cahier des charges :

Donnée entrée par l'utilisateur : précision souhaitée (par exemple 0,1 ou 0,001 ou 0,00005 etc...)
Données affichée : valeur approximative de π à chaque itération.

On utilisera une boucle do...while qui est l'équivalent en C repeter...jusqu'à

On travaillera en double précision (%lf).

On admettra sans le démontrer que la précision sur le calcul de π est égale à 2^* terme.

Exemple : le calcul avec 5 termes donne

$$\pi_{\text{approximatif 5 termes}} = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \right) = 3,339 \quad \text{soit environ } 0,198 \text{ d'erreur et } 1/9 = 0,111..$$



Manipulation n°5

Q.1) Écrire le programme correspondant à l'algorithme donné et au cahier des charges puis le tester en commençant par des valeurs faibles de précision (0,01).
On rappelle $\pi = 3,14159265359....$

Les fonctions

L'utilisation de fonctions en C permet d'alléger l'écriture des programmes, en particulier dans le cas où le même type d'opérations est exécutée plusieurs fois, ceci rend aussi la lecture du programme plus facile.

Un exemple de programme utilisant une fonction est donné à la suite.

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

// declaration des fonctions utilisées
int carre(int nombre);

// fonction main = programme principal
int main()
{
    // variables locales
    int nombreSaisi = 0, CarreNombre = 0;
    printf("Entrez un nombre : ");
    scanf("%d", &nombreSaisi);
    CarreNombre = carre (nombreSaisi); // appel de la fonction
    printf("Le carre de %d est %d\n", nombreSaisi, CarreNombre);

    return 0;
}

// fonction carre
int carre(int nombre) // fonction d'un entier (nombre) qui renvoie un entier (carre)
{
    int c ; // variable locale à la fonction carre
    c = nombre*nombre ; //c = nombre^2
    return c ; // valeur renvoyée par la fonction lorsqu'elle est appelée
}
```

On remarquera que main() est aussi une fonction !

Les fonctions doivent être déclarées avant d'être appelées (utilisées) dans le code.



Travaux dirigés

Q.1) Repérez le programme principal. Où commence-t-il, où se termine-t-il ?

Q.2) Ce programme contient une fonction : quelle est son nom ? Quelle est la variable d'entrée de cette fonction ? Quelle est la variable de sortie ?

Q.3) Comment est transmis le résultat de la fonction ?

Q.4) Que ce passerait-il si on remplaçait la ligne `return c ;` par `return 0 ;` ?

Q.5) Détaillez comment est appelée la fonction.

Q.6) Listez toutes les variables utilisées dans le programme et la fonction et indiquez celles qui sont spécifiques au programme principal et celles qui sont spécifiques à la fonction.

Un capteur de température

Un capteur de température à thermistance fourni une tension VT fonction de la température ambiante. La relation qui permet de déterminer la température en fonction de la tension mesurée est la suivante :

$$T(^{\circ}\text{C}) = a.Vt^3 + b.Vt^2 + c.Vt + d$$

avec $a = -1,78 (^{\circ}\text{C}/\text{V}^3)$; $b = 14,67 (^{\circ}\text{C}/\text{V}^2)$; $c = -56,85 (^{\circ}\text{C}/\text{V})$; $d = 103,26 (^{\circ}\text{C})$
 $0\text{V} < VT < 5\text{V}$



Manipulation n°6

Q.1) Le programme principal est donné, on demande d'écrire la fonction permettant de calculer la température T à partir de la valeur de VT. Cette fonction associée au programme principal permet de calculer toutes les valeurs de température pour VT variant de 0V à 5V par pas de 0,5 V, conformément à la copie d'écran suivante.

```
"C:\Users\Louis\Desktop\C Codeblocks\Projets\exo_fonction\bin\Debug\exo_fo... - [X]
Affichage de temperature
VT = 0.00 V -- temperature = 103.26 oC
VT = 0.50 V -- temperature = 78.28 oC
VT = 1.00 V -- temperature = 59.30 oC
VT = 1.50 V -- temperature = 44.99 oC
VT = 2.00 V -- temperature = 34.00 oC
VT = 2.50 V -- temperature = 25.01 oC
VT = 3.00 V -- temperature = 16.68 oC
VT = 3.50 V -- temperature = 7.68 oC
VT = 4.00 V -- temperature = -3.34 oC
VT = 4.50 V -- temperature = -17.70 oC
VT = 5.00 V -- temperature = -36.74 oC
VT = 5.50 V -- temperature = -61.79 oC

Process returned 0 (0x0)   execution time : 0.273 s
Press any key to continue.
-
```

Programme en page suivante !

Le programme principal est donné :

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

//.....il faudra peut-être rajouter qq chose ici !

// programme principal
int main()
{
    // declaration variables
    float VT = 0 ;
    int i ;

    float T;printf("Affichage de temperature \n\n");
    for (i=0;i<12;i++)
    {
        VT = i*0.5 ;
        T = temperature (VT);
        printf("VT = %.2f V -- temperature = %6.2f oC \n",VT,T);
    }

    return 0;
}
```

Pour ceux qui ont terminé en avance, voici quelques manipulations supplémentaires.



Bonus

Q.1) Écrire un programme qui calcule les solutions d'une équation du second degré. On utilisera une fonction $\text{delta}(a,b,c)$ qui calcule le déterminant $\Delta = b^2 - 4.a.c$. On donnera les solutions réelles et complexes.

Retrouvez d'autres cours et documents sur :

<http://www.louisreynier.com>