



## Structures de données - Les tableaux Analyse descendante - Les fonctions

### Résumé de cours :

Pour la saisie des valeurs entières ou réelles on propose d'utiliser la fonction **scanf** comme le montre le tableau ci-dessous.

Saisie d'un entier	<i>int valNum ; scanf("%d",&amp;valNum) ;</i>
Saisie d'un entier 32 bits	<i>long valLong scanf("%ld",&amp;valNum) ;</i>
Saisie d'un entier 16 bits	<i>short indice ; scanf("%hd",&amp;indice) ;</i>
Saisie d'un réel simple précision	<i>float temperature ; scanf("%f",&amp;temperature) ;</i>
Saisie d'un réel double précision	<i>double population ; scanf("%lf",&amp;population) ;</i>

Pour ce type de saisie le symbole **&** devant la variable est indispensable, il représente l'adresse de la variable. La fonction **scanf** peut ainsi mettre à jour la variable déclarée dans le programme principal. On parle ici de paramètre de sortie.

Pour la mise en forme de l'affichage des nombres avec la fonction **printf**, il est possible de définir la largeur et pour les réels également la précision (nombre de chiffres après la virgule).

#### Exemples :

<i>int valNum = 45; printf("%3d",valNum) ;</i>	<i>float temperature ; printf("%5.2f",temperature) ;</i>
--	--

## Exercice n° 1 : Formatage de l'affichage avec printf et saisie de valeurs avec scanf

---

Soit le programme ci-dessous :

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

int main()
{
    int unEntier = 5;
    float unReel = 10.4 ;
    printf("--- Entier ---\n");
    printf("%d\n%4d\n%04d\n",unEntier,unEntier,unEntier);
    printf("\n--- Réel ---\n");
    printf("%f\n%6.2f\n%.2e\n",unReel,unReel,unReel);
    return 0;
}
```

1. Créez un nouveau projet **td6Ex1** et testez le programme. Expliquez pour chaque élément l'affichage réalisé.

2. A l'aide de la documentation de la fonction printf (lien ci-dessous), modifiez le programme pour qu'il demande une valeur entière à l'utilisateur stockée dans la variable entierEnPlus et affiche cette dernière en décimale et en hexadécimale.

Documentation de la fonction printf : <http://www.cplusplus.com/reference/cstdio/printf/>

## Exercice n°2 : Choisir une opération à partir d'un menu

---

On désire réaliser un programme demandant deux valeurs réelles contenues respectivement dans les variables val1 et val2.

Un menu permet de choisir l'opération à effectuer :

```
1) Faire la somme des valeurs
2) Faire la soustraction
3) Faire le produit
Choisissez l'opération à effectuer
```

La variable resultat contient le résultat de l'opération choisie et est affichée.

Écrivez l'algorithme et codez le programme correspondant à l'énoncé ci-dessus dans le projet **td6Ex2**.

### Exercice n°3 : Affichage de scores

Sous NetBeans, réalisez un nouveau projet nommé **td6Ex3** de type Application en langage C. Codez l'algorithme du programme ci-dessous.

#### Environnement :

En entrée : Le clavier

En sortie : L'écran

#### Schéma algorithmique :

début

Lire : nbMatch

Indice  $\leftarrow$  0

tantque indice < nbMatch

    Lire : scoreDomicile[indice]

    Lire : scoreExterieur[indice]

    Indice  $\leftarrow$  indice + 1

finTantque

pour indice allant de 0 à nbMatch

si scoreDomicile[indice] = scoreExterieur[indice]

alors matchNul  $\leftarrow$  matchNul + 1

sinon si scoreDomicile[indice] = scoreExterieur[indice]

alors gagneDomicile  $\leftarrow$  gagneDomicile + 1

sinon gagneExterieur  $\leftarrow$  gagneExterieur + 1

finSi

finSi

finPour

indice  $\leftarrow$  -1

répéter

    indice  $\leftarrow$  indice + 1

jusqu'à scoreDomicile[indice] = 0 et scoreExterieur[indice] = 0 et indice < nbMatch

si indice  $\neq$  nbMatch

alors Écrire : « il y a eu au moins un match sans aucun but »

finSi

Écrire : « Nombre de match nul : » , matchNul

Écrire : « Nombre de match gagné à domicile : » , gagneDomicile

Écrire : « Nombre de match gagné à l'extérieur : » , gagneExterieur

Fin

#### Lexique :

nbMatch entier

scoreDomicile[15] entier

scoreExterieur[15] entier

matchNul entier

gagneDomicile entier

gagneExterieur entier

Après exécution du programme ci-dessus, l'utilisateur constate des erreurs dans l'affichage du nombre de match nul, gagné à l'extérieur et gagné à domicile. Justifier pourquoi ce programme ne fonctionne pas correctement.

#### **Exercice n°4 : Serrure codée**

---

Une porte à serrure codée permet l'accès à une salle.

Ce code est composé de trois chiffres.

Un programme est chargé de simuler la serrure codée et de vérifier le code saisi parmi 3 codes autorisés contenus dans un tableau nommé codesAutorises.

Dans le cas d'un mauvais code, le programme affichera le message « code incorrect » et redemandera un nouveau code.

L'utilisateur peut se tromper 2 fois, à la troisième fois le programme affichera le message « trop d'erreurs » et se terminera.

Si le code est correct, le message suivant apparaîtra « code OK » et le programme se terminera.

Écrivez l'algorithme et codez le programme correspondant à l'énoncé ci-dessus sous le nom de projet **td6Ex4**.

#### **Exercice n° 5 : Station climatique**

---

On souhaite reprendre la problématique de la station climatique.

On conserve les données de la station climatique (température min et max) sur une semaine.

Plutôt que d'utiliser 2 tableaux distincts, on veut utiliser un seul tableau, mais comprenant plusieurs dimensions.

La première dimension représentera le jour de la semaine et la seconde correspondra au type de température (0 pour min, 1 pour max).

Ainsi, tab[2][0] sera la température mini du 3ième jour et tab[2][1] sera la température maxi du 3ième jour.

■Faites une déclaration de variable pouvant convenir.

■Les données seront saisies au clavier et rangées dans le tableau tabTemp.

■Recherchez le jour du mois où :

- La température min est la plus petite et afficher pour ce jour la température mini et maxi.

- La température max la plus grande et afficher pour ce jour la température mini et maxi.

Écrivez l'algorithme et codez le programme correspondant à l'énoncé ci-dessus sous le nom de projet **td6Ex5**.

## Exercice n° 6 : Sinusoïde

---

On souhaite dessiner une sinusoïde sous forme textuelle.

Pour cela, on va utiliser un tableau de caractère à 2 dimensions qui représentera le plan. La première dimension représentera l'axe des abscisses et sera fixée à 80 et la seconde, l'axe des ordonnées et sera fixé à 40.

Un point sera représenté par le caractère '\*'.

Pour afficher la sinusoïde, il faut procéder en 3 étapes :

1. Initialiser le tableau avec le caractère espace.
2. Remplir le tableau avec le caractère '\*' selon l'algorithme suivant :  
(tabPlan étant notre tableau à 2 dimensions)

```
LARGEUR=80
HAUTEUR=40
rapport= $\frac{LARGEUR}{4\pi}$ 
xReel=0, yReel=0
Faire
    yReel=sin(xReel)
    y= $\frac{HAUTEUR}{2}$ +rapport×yReel
    x=xReel×rapport
    xReel=xReel+0,01
    tabPlan[y][x]='*'
Tant que xReel≤4π
```

La valeur de sin(x) variant entre -1 et +1, on applique un rapport multiplicateur afin que la valeur de sin(x) soit visible. De même, nos indices de tableau ne pouvant pas être négatif, on ajoute HAUTEUR/2 pour la dimension y.

3. Afficher le contenu du tableau.

Codez l'application permettant d'afficher la sinusoïde sous le nom de projet **td6Ex6**.

**Remarque :**  $\pi$  est donné par la bibliothèque <math.h> et se nomme M\_PI.

### **Exercice n° 7 : Fonction addition de nombres réels**

---

On souhaite utiliser un programme qui calcul la somme de 2 valeurs réelles saisies au clavier. Les valeurs seront rangées respectivement dans les variables val1 et val2.

Le calcul de la somme des 2 valeurs sera effectué par une fonction dont le prototype est le suivant :

réel additionReels (réelle v1, réelle v2) ;

Le résultat du calcul sera alors affiché.

1. Sous NetBeans, réalisez un nouveau projet nommé **td6Ex7** de type Application en langage C. A l'aide du document **creationBiblio.pdf**, ajoutez un fichier d'entête **biblioPerso.h** et un fichier source **biblioPerso.c** qui contiendront respectivement les déclarations et les définitions des fonctions.
2. Réalisez l'algorithme correspondant au fonctionnement désiré.
3. Créez la déclaration et la définition de la fonction additionReels.
4. Codez le programme et vérifiez le fonctionnement.

Rappel : Pour pouvoir utiliser une fonction d'une bibliothèque, il faut inclure le fichier de déclaration correspondant (`#include "nomDeBibliotheque.h"`) dans le fichier source.

### **Exercice n° 8 : Fonction calcul du carré d'un entier**

---

On désire réaliser un programme pour calculer le carré d'une valeur entière saisie au clavier et afficher le résultat.

Réalisez un nouveau projet nommé **td6Ex8** de type Application en langage C.

Complétez vos fichiers bibliothèque en créant une fonction nommée `calculerCarre` qui retournera le carré d'une valeur entière passée en paramètre.

Écrivez l'algorithme et codez le programme correspondant à l'énoncé ci-dessus.

## **Exercice n° 9 : Fonction affichage d'un carré**

---

On désire réaliser un programme pour afficher un carré dont la valeur du côté est une valeur entière saisie au clavier.

Réalisez un nouveau projet nommé **td6Ex9** de type Application en langage C.

Complétez vos fichiers bibliothèque en créant une fonction nommée `afficherUnCarre` prenant en paramètre un entier et affichant un carré ayant pour hauteur le nombre passé en paramètre.

Exemple :

`afficherUnCarre(3)` donnera :

```
* * *  
* * *  
* * *
```

Écrivez l'algorithme et codez le programme correspondant à l'énoncé ci-dessus.