

TP n°3 : Lecture, Variables, Sélection simple, Boucle

LS1 : Introduction à l'informatique

1 Exercice: Notion de variable

Synopsis : Il s'agit ici de comprendre la notion de variable et ce que cette notion implique (types, portée, opérations diverses).

Exercice : Ecrire un programme permettant de permuter deux valeurs entières. Réalisez un affichage avant et après permutation.

2. Exercice: utilisation de la structure de sélection simple

Synopsis: il s'agit icic d'utiliser la structure de sélection simple dans un cas concret.

Exercice: vous êtes informaticien et vous travaillez pour les « routes joyeuses de Guadeloupe ». On vous demande de programmer un nouveau système de prévention similaire aux radars déjà installés. L'appareil calcule la vitesse d'un véhicule et affiche les messages suivants pour l'automobiliste en question:

si la vitesse est inférieure strictement à 50 km/h l'appareil affiche: « Attention : votre vitesse trop lente peut provoquer des accidents »

si la vitesse est comprise entre 50 km/h et 90 km/h, l'appareil affiche: « Votre vitesse est bonne: bon voyage! »

si la vitesse est strictement supérieure à 90 km/h, l'appareil affiche : « Danger: vitesse excessive : ralentissez. »

3. Exercice: utilisation de boucle while

Synopsis: utilisation de la boucle sans informations sur le nombre de répétitions

Exercice: On considère l'exercice précédent. Rajouter au début du programme les commandes nécessaires pour que l'ordinateur puisse demander à l'utilisateur de taper une vitesse strictement positive (si l'utilisateur s'amuse à rentrer une valeur négative) jusqu'à ce qu'il rentre une vitesse strictement positive.

4. Exercice: Utilisation de boucle (for ou while)

Soient 10 notes entrées au clavier:

Ecrire un programme qui calcule et affiche la moyenne de ces notes,

5 Exercice : Problème complet

Synopsis : Le but de cet exercice est de comprendre le fonctionnement de la fonction scanf, fonction permettant à l'utilisateur de saisir des données. Pour plus d'informations (entête de la fonction, formats,...) tapez dans un shell

`man scanf`

Cet exercice fait aussi appel à plusieurs autres notions, comme les boucles et les structures conditionnelles.

Exercice : Vous souhaitez vous détendre un peu : vous décidez de programmer un jeu. Le premier joueur saisit un nombre entre 0 et 100. Le deuxième dispose de 5 tentatives pour deviner ce nombre.

Suivant la valeur absolue de la différence entre l'essai du second joueur et le nombre choisi, vous afficherez les messages suivants :

- Différence inférieure à 5 : « Tu brûles !!! »
- Différence comprise entre 5 et 20 : « Tu chauffes !!! »
- Différence comprise entre 20 et 30 : « C'est tiède ... »
- Différence supérieure à 30 : « C'est froid »

L'exercice peut-être traité en suivant les étapes proposées ci-dessous :

2.1 Partie saisie - 1

Implémentez uniquement la saisie du joueur 1 et l'affichage du nombre saisi

2.2 Partie saisie -2

Complétez la partie précédente en ajoutant la saisie du joueur 2 et en comparant le nombre nouvellement saisi avec celui saisi par le joueur 1.

Si celui-ci est supérieur, affichez : « C'était plus petit !! » sinon « C'était plus grand » avant d'afficher finalement la valeur saisi par le joueur 1

2.3 Boucles

Utiliser une boucle parmi celles vues en cours pour permettre de répéter l'opération précédente un nombre fini de fois.

2.4 Finalisation

Compléter le code déjà produit et terminer la réalisation de l'exercice.

1) Fonction scanf

#include<stdio.h>

Entête : int scanf (const char *format, ...);

Utilisation : scanf sert à demander à l'utilisateur de rentrer au clavier une variable donnée (indiquée par son adresse) en utilisant un format d'entrée défini par le programmeur.

Type de données :

Type de la variable	Séquence
Caractère (char)	%c
Entier (int)	%d
Réel (float)	%f
Tableau de caractères (char*)	%s
Entier long (long)	%ld
Réel long (double)	%f

Exemple :

`scanf(''%d'', &a);` => Enregistre dans la variable **a** l'entier entré au clavier par l'utilisateur.

Attention toutefois au type des données saisies

2) Structure de sélection simple

2.1) if ... else

Structure conditionnelle permettant d'effectuer une ou plusieurs actions à la suite si une condition est vérifiée (ou pas).

Ex :

N°	Algorithme	Langage C
1	Si A>4 faire	if (A>4)
2	A ← 1	{ A=1; }
3	Sinon	else
4	A ← A + 1	{A=A+1; }
5	FinSi	

3) Boucles

3.1) for

Structure de programmation permettant de répéter une ou plusieurs instructions un nombre fini de fois. Se base sur l'utilisation d'un compteur incrémenté à chaque tour.

Exemple d'utilisation :

```
for (i=0 ; i<k ; i++)  
{  
    <suite d'actions>  
}
```

3.2) while

Structure de programmation permettant de répéter une ou plusieurs instructions jusqu'à ce que la condition de départ ne soit plus remplie

Exemple d'utilisation

```
while ( <condition> )  
{  
    <suite d'actions>  
}
```