

# Cahier de TD n° 1

**THEME 1 : Stockages et types de variables, affectations, expressions**

Identifiants

Variables et initialisations

Variables et types

Calculs simples

**THEME 2 : tests, conditions, sélection**

Test simple, si...alors...sinon

Test if...else if...else

Sélection switch...case

**THEME 3 : boucle while et do while**

**THEME 4 : boucle for**

[illegible]

## Variables et types :

remplissez le tableau suivant avec les valeurs prises par les variables

```
int main()
{
    long    val_1, val_2;
    double  x_1, x_2, y_1;
}
```

	val_1	val_2	x_1	x_2	y_1
val_1 = 11;					
x_1 = 4.0E+2;					
val_2 = val_1 - 6;					
y_1 = 3 * x_1 - 200.0;					
x_2 = val_1 * val_2;					
x_1 = val_1 / val_2;					
y_1 = x_2 - val_1					
x_2 = val_2 / val_1					
val_2 = x_1;					

## Calculs simples :

- Ecrivez un programme qui calcule et affiche la moyenne de trois nombres entiers saisis au clavier
- Ecrivez un programme qui calcule et affiche l'écart-type de trois nombre entiers saisis au clavier.

Formule de l'écart-type :  $\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{m})^2$ , où n est le nombre de valeurs, ici n=3.

$\bar{m}$  est la moyenne des valeurs.

- Ecrire un programme qui calcule la différence, en secondes, entre deux dates (en heures, minutes, secondes) de la même journée.
- Ecrivez un programme de suivi d'épargne : un capital  $C$  est placé sur un compte dont le taux d'intérêt semestriel  $T$  est de 2,43 %. Le programme doit calculer et afficher le capital au bout de : 6 mois, 12 mois, 18 mois et 30 mois.

## **THEME 2 : tests, conditions, sélection**

Rappel : lorsqu'on propose une saisie dans un programme, il faut systématiquement afficher un message pour indiquer à l'utilisateur ce qu'il doit saisir !

### **Test simple, si...alors...sinon**

- **Analyse d'un programme simple**

```
int main()
{
    double borne_basse, borne_haute;
    double valeur;
    long test_bas, test_haut;

    borne_basse = -3.2;
    borne_haute = 5.56;

    printf("saisissez une valeur reelle :");
    scanf("%lf",&valeur);

    test_bas = (valeur < borne_basse);
    test_haut = (valeur > borne_haute);

    if (test_bas == 1)
    {
        printf("la valeur est en dessous de l'intervalle");
    }
    if (test_haut == 1)
    {
        printf("la valeur est au dessus de l'intervalle");
    }
    if ((test_bas == 0) && (test_haut == 0))
    {
        printf("la valeur est dans l'intervalle");
    }
}
```

Les variables test\_bas et test\_haut peuvent-elles prendre d'autres valeurs que 0 et 1 ?

Justifiez votre réponse.

Combien de messages le programme peut-il afficher lors d'une exécution ?

Expliquez pourquoi.

Indiquez les valeurs des variables et des conditions au cours du déroulement du programme lorsque l'utilisateur saisit :

- a) 0.05
- b) 6.33

- Ecrire un programme qui fait : la saisie de trois niveaux de pollution, exprimés en microgrammes par m<sup>3</sup> (µg/m<sup>3</sup>), qui calcule la moyenne de ces valeurs, et indique, à l'aide d'un message, si un seuil de 480 µg/m<sup>3</sup> est dépassé, auquel cas le niveau est considéré comme dangereux. Les données sont des nombres à virgule (type réel);
- Ecrire un programme qui affiche une question et 4 possibilités de réponses, chacune précédée d'un numéro entre 1 et 4. L'utilisateur doit sélectionner le numéro de la réponse choisie, et le programme doit indiquer s'il s'agit de la bonne réponse ou si c'est une erreur.

### Test if...elseif...else

- Reprenez le programme `intervalle` de la section précédente avec une ou des structures `si...alors...sinon si`. Indiquez pourquoi cette structure est bien adaptée au problème traité.
- Ecrire un programme qui indique si une année est bissextile ou non : une année multiple de 4 est en général bissextile. Attention toutefois, les années séculaires (multiples de 100) ne sont pas bissextiles, sauf si elles sont multiples de 400 ! (1400 et 1900 ne sont pas bissextiles, mais 1200 et 2000 le sont).
- Ecrire un programme qui saisit un poids en kg, une taille en m et un âge; qui calcule un coefficient de poids  $C = \text{poids} / \text{taille}^2$ ; puis en fonction de l'âge, qui indique si la personne est en surpoids selon le barème suivant :

Tranche d'âge	Seuil de surpoids
16 ans et moins	$C > 27,3$
De 17 à 48 ans	$C > 23$
49 ans et plus	$C > 21,5$

- Ecrire un programme qui indique les services disponibles sur une ligne ADSL en fonction de la distance au NRA (Nœud de Raccordement Abonné). Il faut saisir la distance D en mètre entre l'habitation et le NRA, puis calculer l'atténuation théorique A (en db) en utilisant la formule :  $A = D / 68$ ;

Atténuation théorique (en db)	Services ADSL
$A < 37$	Internet 5 Mb; téléphonie IP; TV
$37 \leq A < 56$	Internet 5 Mb; téléphonie IP
$A \geq 56$	Internet 2 Mb

## Sélection switch...case

- Dans la cuisine d'un restaurant où vous êtes en stage, on dispose d'un four à 5 niveaux de thermostat (numérotés 4,5,6,7,8) , et de 6 plats à faire cuire : un pavé de saumon, un gigot d'agneau piqué à l'ail, une tarte à la quetsche, un poulet en croûte de sel, un canard laqué et un gratin de légumes au parmesan. Le chef de cuisine a mis le four à préchauffer avec un certain thermostat, et votre programme doit, à partir de cette valeur, afficher le plat à mettre au four avec le temps de cuisson. Vous savez en outre que :
  - Le saumon est cuit en 8 minutes au thermostat 4,
  - Le gigot se cuit au thermostat 7 en 25 minutes, et à la même température, une tarte cuit en 35 minutes;
  - Les volailles se cuisent à la température la plus chaude, le poulet en 40 minutes, le canard en une heure;
  - Le thermostat 5 convient parfaitement pour les légumes, 15 minutes suffisent.Ecrivez le programme correspondant, bon appétit !
- On veut maintenant améliorer le programme concernant les services ADSL en proposant à l'abonné de choisir les services qu'il veut parmi ceux qui sont disponibles en considérant que son atténuation lui permet d'accéder à tous les services (cas où  $A < 37$ ). L'internet 5 Mb vaut 19,99 €/mois, la TV 4,99 € par mois, la téléphonie IP 6 € / mois. Ecrire un programme qui permet de choisir parmi les options disponibles et qui indique le montant à payer par mois.
- Enfin, le FAI (Fournisseur d'Accès Internet) propose les remises suivantes : 5% de remise pour un engagement de 12 mois, 12 % pour un engagement de 24 mois. Modifiez le programme pour que l'utilisateur saisisse le nombre de mois d'engagement, le programme indiquera : le montant de chaque mensualité, et le montant total sur toute la durée de l'engagement.

### **THEME 3 : boucle while et do while**

Saisie sécurisée : pour rendre les programmes plus fiables et sécurisés, on peut contrôler que les saisies sont conformes à ce qu'on attend avant de continuer le programme.

- Ecrire un programme qui calcule la racine carrée d'une valeur à virgule :  
il faut vérifier que la valeur saisie est positive avant de faire le calcul : on utilisera donc une boucle while ou do...while.

Pour le calcul de la racine carrée, on utilisera la fonction sqrt(). Attention à bien choisir la boucle : while, ou do...while ?

- Ecrire un programme auquel on fournit un rang entier  $p$ , et qui calcule la valeur des suites suivantes au rang  $p$  :

$$\text{Suite } U_n : \begin{cases} U_0 = 4,5 \\ U_{n+1} = 2 \cdot U_n - 3; \end{cases}$$

$$\text{Suite } V_n : \begin{cases} V_0 = 6 \\ V_{n+1} = V_n/2 - 2.5; \end{cases}$$

$$\text{Suites } W_n \text{ et } X_n : \begin{cases} W_0 = 2; X_0 = -1 \\ W_{n+1} = W_n + X_n; \\ X_{n+1} = X_n - W_n; \end{cases}$$

$$\text{Suite } Y_n : \begin{cases} Y_0 = 4; Y_1 = 2 \\ Y_{n+1} = Y_n/Y_{n-1} + 1; \end{cases}$$



- Le nombre mystère :

On cherche à faire deviner à l'utilisateur un nombre, compris entre 1 et 10 000, choisi au hasard par l'ordinateur. L'utilisateur procède par essais successifs : à chaque essai, l'ordinateur indique si le nombre rentré est plus grand ou plus petit que le nombre mystère. Le jeu continue jusqu'à ce que l'utilisateur trouve le nombre mystère.

Pour faire un tirage de nombre au hasard, on utilisera l'instruction : `rand( )`; cette instruction donne une valeur entière comprise entre 0 et un très grand entier nommé `RAND_MAX`. Vous pouvez utiliser l'opérateur `%` (modulo).

- Méthode de Newton : il existe une méthode simple pour calculer la racine carrée d'un nombre  $x$  à l'aide d'une suite :

La suite :  $U_0 = x$ ;  $U_{n+1} = U_n/2 - x/(2.U_n)$  converge vers  $\sqrt{x}$  (c'est à dire que la valeur de  $U_n$  se rapproche de plus en plus de  $\sqrt{x}$  quand  $n$  augmente). Ecrire un programme qui calcule la racine d'un nombre  $x$  et qui s'arrête lorsque la précision obtenue est de  $10^{-8}$ .

- Ecrire un programme de gestion de compte d'épargne :

On dépose un montant initial de  $M$  €, et tous deux mois, un montant  $m$ . Tous les 6 mois, les intérêts, à un taux  $T$ , sont versés sur le compte. Tous les ans, si le montant dépasse un seuil  $S$ , une taxe à un taux  $T_{taxe}$  est prélevé. Egalement tous les ans, des frais de gestion de compte, d'un montant de 28,75 €, sont prélevés.

Ecrire un programme qui saisit le montant initial  $M$  et le montant du versement  $m$ , et qui affiche le solde du compte par périodes de deux mois. On affiche aussi les autres évènements : versement des intérêts, prélèvements.

On arrête l'affichage : au bout de 4 ans, ou si le solde du compte atteint ou dépasse un plafond  $P$ .

Données :

Le montant  $M$  doit être de 15 € minimum, le montant  $m$  de 150 € minimum, le taux  $T$  est de 4,5 %, le taux  $T_{taxe}$  de 1,023 %. Le seuil  $S$  est de 6 000 €, le plafond  $P$  de 12 000 €.

## **THEME 4 : boucle for**

- Reprendre l'exercice sur les suites abordés dans le thème précédent : choisissez deux suites parmi celles proposées et modifiez le programme existant afin de le réaliser avec un boucle pour. Peut-on utiliser une boucle for à la méthode de Newton vue dans le thème précédent ?
- Ecrire un programme qui calcule et affiche les carrés des 25 premiers entiers.
- Nombres premiers. Il existe une méthode simple (mais très peu efficace) permettant de déterminer les nombres premiers. Un nombre premier est un nombre qui n'est divisible que par 1 et par lui même (1 n'est pas un nombre premier selon cette définition). Pour déterminer si un nombre est premier, il suffit de voir s'il est divisible par un nombre qui est plus petit que lui. S'il est divisible, il n'est pas premier.

Ecrire un programme qui indique si un nombre entier saisi au clavier est un nombre premier. Indication sur la méthode : on utilisera une variable dont le seul but est de stocker l'information 'le nombre est premier'. On pourra utiliser un nombre entier qui vaudra 1 si le nombre est premier, et qui vaudra 0 sinon. Au départ, cette variable vaudra 1 (on suppose donc que le nombre est premier)

- A partir du programme précédent, faire un programme qui fait la liste des nombres de 1 à 1000, et indique si chacun de ces nombres est premier ou non.
- Ecrire un programme qui affiche les lettres de l'alphabet, en majuscule, et leur code de la table ASCII, puis qui fait de même avec les minuscules.
- Ecrire un programme qui saisit deux nombres à virgule, qui sont les deux bornes d'un intervalle, et un nombre entier : ce nombre entier est le nombre de points, régulièrement espacés, que l'on veut placer dans cet intervalle (par exemple, quand on veut faire le graphe d'une fonction avec un ordinateur, on demande, dans l'intervalle  $[-1,5; +2]$  de tracer 1000 points. Le programme devra afficher la coordonnée de chaque point.