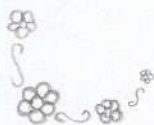


Cahier de TD n° 1



SEANCE 1 : STOCKAGES ET TYPES DE VARIABLES, AFFECTATIONS, EXPRESSIONS 1

IDENTIFIANTS :	1
VARIABLES ET INITIALISATIONS :	1
CALCULS SIMPLES :	1

SEANCE 2 : TESTS, CONDITIONS, SELECTION 2

TEST SIMPLE, SI...ALORS...SINON.....	2
TEST SI...ALORS...SINON SI.....	3

SEANCE 3 : BOUCLE TANT QUE ET FAIRE...TANT QUE..... 4**SEANCE 4 : BOUCLE POUR 6****EXERCICE DE SYNTHÈSE 7**

Séance 1 : Stockages et types de variables, affectations, expressions

Identifiants :

Indiquez si les noms de variables suivants sont corrects, et sinon, pour quelle raison

1) entier ✓
 un=entier ✗
 un_entier ✓
 x_435 ✓
 X435 ✓
 a' ✗
 6o6on ✗
 identificateur_de_variable ✓
 CoMPTeuR ✓

Variables et initialisations :

remplissez le tableau suivant avec les valeurs prises par les variables

programme calculs;

entier a, b, compteur, limite;
 réel x, deriv, vitesse;
 caractère lettre, let2;

	a	b	compteur	limite	x	deriv	vitesse	lettre	let2
a ← -14;	-14	17	0	34	6.022E+23	168616000	6077000	83	
lettre ← 'a'+a;	-14	17	0	34	6.022E+23	168616000	6077000	83	
b ← 3-a;	-14	17	0	34	6.022E+23	168616000	6077000	83	
compteur ← 0;	-14	17	0	34	6.022E+23	168616000	6077000	83	
limite ← b+b;	-14	17	0	34	6.022E+23	168616000	6077000	83	
b ← b-1;	-14	16	0	34	6.022E+23	168616000	6077000	83	
x ← 6.022E+23;	-14	16	0	34	6.022E+23	168616000	6077000	83	
vitesse ← x * 1.0E-17;	-14	16	0	34	6.022E+23	168616000	6077000	83	
deriv ← vitesse*(b-a);	-14	16	0	34	6.022E+23	168616000	6077000	83	
compteur ← compteur % limite;	-14	16	0	34	6.022E+23	168616000	6077000	83	

Calculs simples :

- Ecrivez un programme qui calcule et affiche la moyenne de trois nombres entiers saisis au clavier

- 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.
- On lâche un caillou dans un puits asséché et très sombre. On entend un bruit d'impact au bout de T secondes. Ecrire un programme qui calcule et affiche : la profondeur du puits et la vitesse du caillou à l'impact.

Séance 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

programme intervalles;

réel borne_basse, borne_haute;
réel valeur;
entier test_bas, test_haut;

borne_basse \leftarrow -3.2;
borne_haute \leftarrow 5.56;

afficher("saisissez une valeur réelle :");
saisir(valeur);

test_bas \leftarrow (valeur < borne_basse);
test_haut \leftarrow (valeur > borne_haute);

```
si (test_bas = 1) alors
{
    afficher("la valeur est en dessous de l'intervalle");
}
si (test_haut = 1) alors
{
    afficher("la valeur est au dessus de l'intervalle");
}
si ((test_bas = 0) ET (test_haut = 0)) alors
{
    afficher("la valeur est dans l'intervalle");
}
```


1 les variables `test_bas` et `test_haut` peuvent-elles prendre d'autres valeurs que 0 et 1 ? justifiez votre réponse.

2 Combien de messages le programme peut-il afficher lors d'une exécution ? Expliquez pourquoi.

3 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 m³ ($\mu\text{g}/\text{m}^3$), qui calcule la moyenne de ces valeurs, et indique, à l'aide d'un message, si un seuil de 480 $\mu\text{g}/\text{m}^3$ 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 si...alors...sinon si

- 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	$C > 21,5$

Séance 3 : boucle tant que et faire...tant que

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 tant...que ou faire ... tant que.

Pour le calcul de la racine carrée, on utilisera l'instruction `racine()`. Attention à bien choisir la boucle : tant que, ou faire...tant que ?

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

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

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

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

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

$$U_0 \times q^m \pm k^m$$

- 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 : `hasard()`; 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 événements : 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 €.

Séance 4 : boucle pour

- 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 pour 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.

Exercice de synthèse

- Ecrire un programme qui saisit un texte caractère par caractère jusqu'à ce que l'utilisateur entre le caractère '#'. A chaque caractère tapé, le programme doit indiquer s'il s'agit d'un chiffre, d'une lettre de l'alphabet, ou d'un autre caractère. S'il s'agit d'une lettre de l'alphabet, le programme doit afficher cette lettre en minuscule.