

Exercice 1 :

Déclarations (constantes, variables), types simples, code ASCII, commentaires, opérateurs arithmétiques binaires et unaires, opérateurs d'affectation, conversions.

- 1.1.** Ecrire un programme C qui lit deux réels, les affiche, calcule leur somme et affiche le résultat.
- 1.2.** Ecrire un programme C qui permute les valeurs de deux variables lues au clavier.
- 1.3.** Ecrire un programme C qui calcule le pourcentage x d'un nombre y. Ce nombre ainsi que le pourcentage sont saisis au clavier.
- 1.4.** Ecrire un programme C qui donne une température en degré Celsius à partir d'une température Fahrenheit ($C = (5/9) * (F - 32)$).
- 1.5.** Ecrire un programme C qui lit un entier et l'affiche en décimal, en octal et en hexadécimal.
- 1.6.** Ecrire un programme C qui lit un entier et affiche à l'écran si c'est 0, un nombre pair ou un nombre impair.
- 1.7.** Ecrire un programme C qui déclare un caractère (type char) et l'initialise une première fois avec le caractère '5', puis avec le caractère correspondant au code ASCII 96.

Exercice 2 :

Instructions de contrôle

- 2.1.** Ecrire un programme C qui teste si un caractère saisi au clavier est un chiffre ou si c'est un caractère compris entre 'a' et 'z' ou encore entre 'A' et 'Z'.
- 2.2.** Ecrire un programme C qui lit une suite d'entiers strictement positifs et qui s'arrête si on saisit la valeur -1.
- 2.3.** Ecrire un programme C qui demande "Quelle table de multiplication voulez-vous, tapez 0 (zéro) pour sortir ?", qui saisit un caractère, si le caractère est compris entre '1' et '9' alors on fera afficher la table correspondante puis on réitérera le processus, sinon on affichera "ce n'est pas dans les possibilités du programme, recommencez !" et on réitérera le processus.
- 2.4.** Ecrire un programme C en utilisant l'instruction Switch qui lit un caractère (fonction getchar()), qui affiche "un" si le caractère '1' a été saisi, "deux" si c'est le caractère '2', "trois" si c'est '3' et "autre caractère" sinon.

Exercice 3 :

Tableaux

- 3.1.** Déclarer un tableau de 10 entiers, une variable de contrôle de boucle de type entier; puis réaliser l'initialisation du tableau avec la valeur 0 à l'aide d'une boucle.
- 3.2.** Déclarer un tableau de 5 entiers en l'initialisant à la déclaration avec les valeurs décroissantes allant de 4 à 0.
- 3.3.** Déclarer un tableau de dimension 2, de trois lignes et 4 colonnes. Initialiser les cases du tableau avec des valeurs allant de 12 à 23, la première ligne contiendra les valeurs 12, 13, 14, 15; la deuxième 16, 17, 18, 19; la troisième 20, 21, 22, 23; afficher les cases du tableau en utilisant la notation indicée.
- 3.4.** Faire le calcul de multiplication d'une matrice (M lignes, L colonnes) par une matrice (L, N) : résultat (M,N).

Exercice 4 :

Opérateurs "adresse de" et "contenu de", appel de fonction par référence

- 4.1.** Ecrire une fonction qui permute les valeurs de ses deux paramètres. Tester cette fonction par un appel depuis votre programme principal.
- 4.2.** Pourquoi dans la question 4.1 utilise-t-on des pointeurs ? Aurait-on pu utiliser des entiers ? Justifier.
- 4.3.** Ecrire un programme qui déclare deux variables i et j; la première (i) sera une variable de type entier, la seconde (j) sera une variable de type pointeur sur entier. C'est-à-dire que j contiendra une valeur qui sera l'adresse (emplacement mémoire) d'un entier. Affecter à l'entier i la valeur 5, affecter à l'adresse j l'adresse de l'entier i, afficher l'entier i et le contenu de ce qui est pointé par la variable j.
- 4.4.** Modifier le programme précédent en ajoutant à la fin les instructions suivantes : incrémenter la donnée pointée par la variable j, afficher i, multiplier i par 5, afficher la donnée pointée par j, incrémenter l'adresse j et afficher de nouveau la donnée pointée par j.
- 4.5.** Reprendre l'exercice 3.3 avec la notation en pointeur pour la ligne.
- 4.6.** Reprendre l'exercice 3.3 avec une notation en pointeur uniquement.
- 4.7.** Ecrire un programme qui lit une liste de Nb nombres, la décale d'un cran vers le haut (le premier doit se retrouver en dernier), l'affiche puis la décale vers le bas. Les nombres doivent être stockés dans un tableau avec une notation en pointeurs.

Exercice 5 :

Chaines de caractères

5.1. Ecrire un programme C qui détermine le nombre et la position d'une sous-chaîne dans une chaîne.

5.2. Ecrire un programme C qui transforme tous les caractères minuscules d'une chaîne en caractères majuscules.

5.3. Ecrire un programme C qui permet de jouer au jeu du pendu. Le joueur 1 entre le mot, l'ordinateur l'enregistre et affiche une série de lignes blanches pour que le joueur 2 ne puisse pas lire le mot, il affiche ensuite une série d'étoiles qui correspond au nombre de lettres du mot à trouver. Le joueur 2 propose des caractères jusqu'à ce qu'il ait trouvé le mot, ou qu'il ait perdu (nombre de coups > 10). À chaque fois que le joueur 2 propose un caractère l'ordinateur affiche le mot avec des * et les caractères déjà trouvés.

Exemple :

Entrez un mot : bonjour

caractere ? a

caractere ? o

*o**o**

caractere ? u

*o**ou*

caractere ? n

*on*ou*

caractere ? j

onjou

caractere ? s

onjou

caractere ? b

bonjou*

caractere ? r

bonjour

VOUS AVEZ GAGNE