Vous aurez besoin de 4 copies : une par partie

Durée 2 heures - Sans documents (sauf diagrammes de C), pas de calculette ni de traducteur électronique.

Partie 1 (5 points) → copie n°1

1. Apprentissage de la multiplication pour les petits

On se propose d'écrire un programme permettant l'apprentissage de la multiplication pour des élèves de CE2. Pour cela on demandera à l'élève deux nombres entre 1 et 9 (que l'on vérifiera) et ensuite, on affichera sous forme d'étoiles la multiplication comme le montre l'exemple suivant :

L'élève rentre 3 et 5, il affiche 3 x 5 = 15 -> 3 fois 5 étoiles = 15 étoiles.

Entre deux nombres STP : 3 5
3 x 5 = 15 -> 3 x *** = ***** + ***** + ***** = ***********

1.1 Ecrire le programme C correspondant.

1.2 Suivant le même principe, afficher la table de multiplication d'un nombre demandé à l'élève.

Entre un nombre STP : 4
4 x 1 = 4 -> 4 x * = * + * + * + * = **
4 x 2 = 8 -> 4 x ** = ** + ** + ** + ** = ******
4 x 3 = 12 -> 4 x * = *** + *** + *** + *** = ************
etc. (jusqu'à 4 x 9)

Partie 2 (5 points) → copie n°2

2. Cryptographie

On souhaite réaliser un programme permettant de coder une phrase saisie par l'utilisateur avec un algorithme très simplifié de cryptographie. Chaque caractère de la phrase sera décalé vers la droite de D positions dans l'alphabet.

Pour rendre l'algorithme un peu plus sûr, ce décalage D changera pour chaque caractère de la phrase et sera calculé par un générateur pseudo-aléatoire dit de "Fibonacci". Le décalage D sera donc donné par la formule suivante $X_n = (X_{n-1} + X_{n-2})$ modulo 26.

L'utilisateur entrera au début du programme, 2 nombres entiers (X_0 et X_1) qui serviront à initialiser cette suite. Ces 2 nombres constituent la clé secrète qui permettra de déchiffrer ultérieurement le message (le programme de déchiffrement n'est pas demandé). Le premier caractère significatif sera donc codé avec X_2, le second avec X_3, etc.

Consignes:

- la phrase sera lue caractère par caractère (avec getchar()), elle se termine par un '.' ;
- les caractères autres que les lettres de 'a' à 'z' seront ignorés ;
- les caractères seront convertis en minuscules au moyen de la fonction tolower() décrite ci-dessous ;
- si le décalage fait "sortir" de l'alphabet, vous devez en tenir compte.
  Par exemple, la lettre 'v' avec un décalage de 8 donnera 'd'.

Remarques :

- Vous pouvez utiliser des expressions sur les caractères, on a notamment : 'c' - 'a' = 2, 'v' + 4 = 'z'…
- Vous devez utiliser la fonction prédéfinie *char tolower(char)* qui retourne l'équivalent en minuscule du caractère alphabétique passé en paramètre.

Exemple:

Entrez X0 et X1 : 3 4
Entrez le message initial : test.
apkw

Partie 3 (5 points) → copie n°3

3.1 Encore la multiplication, mais pour les plus grands

On voudrait réaliser une multiplication en utilisant uniquement des additions, soustractions, et des multiplications ou divisions par 2. L'objectif est de répéter ces opérations de base en diminuant à chaque étape la valeur du multiplicateur b selon qu'il est pair ou impair :

a * 0 = 0	si b=0
a * b = a * (b-1) + a	si b est impair
a * b = (2 * a) * (b/2)	si b est pair et différent de 0

Ecrire un programme qui lit deux entiers positifs, et affiche leur produit selon le principe ci-dessus. Afin de converger plus rapidement, le programme fera en sorte que le premier multiplicateur (a) soit plus grand que le second (b). On affichera toutes les étapes de calcul telles qu'elles figurent dans l'exemple ci-dessous :

Exemple :

Entrez 2 nombres entiers : 7 36

36 * 7 = 36 * 6 + 36
= 72 * 3 + 36
= 72 * 2 + 108
= 144 * 1 + 108
= 144 * 0 + 252
= 252

3.2 Qu'affiche ce programme ?

```
{  int x, y, i;
   int *z,*v;
   x = 7;
   y = 4;
   z = &y;
   printf("P1 : x = %d, y = %d, z = %d \n", x, y,*z);
   *z = (x++);
   v = &x;
   (*z)--;
   printf("P2 : x = %d, y = %d, z = %d, v = %d \n", x+2, y ,*z,*v);
   y = 0 ;
   for( i = 1; i <= *v; i++)
      y+= i;
   y/= *v - 1;
   z = &i;
   printf("P3 : x = %d, y = %d, z = %d, v = %d \n", x, y,*z,*v);
    }
```

Partie 4 (5 points) → copie n°4

4. Elections présidentielles

On voudrait écrire un programme qui calcule les résultats du premier tour d'une élection à vote majoritaire. On suppose que 4 candidats se présentent. Chaque votant vote pour zéro (vote nul) ou un seul candidat, les candidats étant numérotés de 1 à 4.

On voudrait dans un premier temps 1) saisir les résultats bulletin par bulletin, 2) à l'issue de la saisie, afficher le nombre de votants et le nombre de bulletins exprimés (non nuls) et 3) afficher les pourcentages obtenus par chaque candidat.

4.1 Ecrire le programme C qui demande à chaque votant son choix (un des candidats ou un vote nul) et, quand il n'y a plus de votant, affiche les résultats demandés (le nombre de votants n'est pas connu au départ).

Dans un second temps, on voudrait afficher le résultat final à savoir : si un candidat a la majorité absolue (>50% voix) alors il est élu, sinon les candidats ayant les 2 meilleurs scores accèdent au 2d tour.

4.2 Continuer le programme précédent pour afficher le résultat final : les candidats qui passent au second tour ou le cas échéant, le candidat élu dès le premier tour.