

Exercice 1 :

```
1  #include<stdlib.h>
2  #include<stdio.h>
3  int main() {
4      short tab[10] = { 2 };
5
6      printf("tab[0] : %d \n", tab[0]);
7      printf("tab[1] : %d \n", tab[1]);
8      printf("taille : %d \n", sizeof(tab));
9
10     for (int i = 3; i <= 8; i++) {
11         tab[i] = 2 * i;
12     }
13
14     int i = 0;
15     while (i < 10 && tab[i] < 7) {
16         printf("%d : %d\n", i, tab[i]);
17         i++;
18     }
19     tab[0] = tab[i];
20
21     return (EXIT_SUCCESS);
22 }
```

- 1) Combien de cases le tableau tab comporte-t-il ?
- 2) Donnez l’affichage des lignes 6 - 7 - 8.
- 3) Donnez le contenu de toutes les cases du tableau lors du passage à la ligne 13.
- 4) Donnez l’affichage produit par la ligne 16.
- 5) Donnez le contenu de toutes les cases du tableau lors du passage à la ligne 20.

Exercice 2 : Chaine de caractères

- 1) Définir une chaine de caractères aléatoires de taille n contenant les lettres d’alphabet en minuscule (sachant que le code ASCII de ‘a’ est 97)
$$\text{MIN} + (\text{rand}() \% (\text{MAX} - \text{MIN} + 1))$$
- 2) Ecrire un code calculant le nombre d’occurrences de la lettre ‘e’.
- 3) Compléter votre code afin de calculer le nombre d’occurrence de toutes les lettres d’alphabet. Afficher le résultat.

Exercice 3 – Remplacement des doublons

Soit un tableau T contenant des entiers strictement positifs. Ecrire l'algorithme qui remplace tous les doublons par des 0. Exemple : le tableau 1,3,3,1,2,5,1 deviendra 1,3,0,0,2,5,0.

Exercice 4 – Compression

Nous souhaitons écrire un programme qui compresse sans perte des séquences de 0 et de 1.

Les données brutes sont une suite de 0 et de 1, terminée **systématiquement** par la valeur -1 exemple :

0 0 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 -1

L'algorithme de compression : on recherche les groupes d'éléments successifs identiques (par exemple, 3 fois de suite la valeur "1"). Lorsqu'un élément apparaît une seule fois, il est conservé tel quel. Mais s'il apparaît n fois de suite avec $n \geq 2$, il est stocké sous la forme " n suivi de l'élément répété". Par exemple, la séquence ci-dessus sera compressée en :

2 0 4 1 0 3 1 2 0 1 4 0 -1

