

2 h - Sans documents (sauf diagrammes de C) ni calculatrice

Vous aurez besoin de 4 copies : une par exercice

**1 Détecteur d'antanaclases (4 points)**

Ecrire la fonction d'entête : `int trouver(char* str, char* m, int* rep, int n)` qui permet de trouver les différentes positions d'un mot dans une chaîne de caractères.

- Le paramètre `str` est l'adresse du premier élément de la chaîne de caractères dans laquelle il faut chercher le mot.
- Le paramètre `m` est l'adresse du premier élément de la chaîne de caractères représentant le mot à rechercher.
- Le paramètre `rep` est l'adresse du premier élément d'un tableau d'entiers dans lequel seront reportées les différentes positions du mot `m` dans la chaîne `str` (c'est à dire les indices où le mot `m` commence dans `str`).
- Le paramètre `n` indique la taille de ce tableau d'entiers. Ainsi, la fonction ne pourra reporter au maximum que les `n` premières positions du mot `m` dans la chaîne `str`.
- La valeur retournée par la fonction sera le nombre de fois où le mot `m` a été trouvé dans la chaîne `str`.

Exemple d'utilisation :

```
void main()
{
    char texte[]="Le coeur a ses raisons que la raison ne connaît pas.";
    char mot[]="raison";
    int pos[5];
    int i,nb=trouver(texte,mot,pos,5);
    printf("le mot %s a été trouvé au(x) position(s): ", mot);
    for(i=0; i<nb; i++) printf("%d ",pos[i]);
    printf("\n");
}
```

Affichage provoqué : le mot raison a été trouvé au(x) position(s): 15 30

Nouvelle copie

**2 Station Météo (6 points)**

Dans une station météo, on relève chaque jour la température minimum et la température maximum. A la fin du mois, on saisit les données dans une structure de type `unReleveMensuel`.

Le relevé mensuel contiendra les informations suivantes :

- nom de la station (chaînes de caractères de 20 caractères maximum)
- coordonnées GPS (latitude, longitude, altitude)
- mois
- tableau des températures min en degré (taille max 31)
- tableau des températures max en degré (taille max 31)

D'autre part, on supposera connue la fonction `int nbDeJours(int mois)` qui retourne le nombre de jours que contient le mois passé en paramètre.

2.1 Ecrire le type `uneCoordGPS` comprenant trois réels, en degré pour la latitude et la longitude et en mètre pour l'altitude. Ecrire le type `unReleveMensuel` comprenant les champs suivants : le nom de la station, les coordonnées GPS (`uneCoordGPS`), le mois concerné (sous forme d'entier compris entre 1 et 12), le tableau des températures journalières minimum en degrés et le tableau des températures journalières maximum en degrés.

2.2 Ecrire la fonction `initUnReleveMensuel()` qui initialise un relevé à partir d'un dialogue avec l'opérateur, le relevé sera passé en argument de sortie de la fonction et le mois en argument d'entrée.

2.3 Ecrire une fonction `stockDansFichier()` qui écrit dans un fichier `releves.dat` tous les relevés demandés à un opérateur pour des stations différentes (30) mais pour un mois précis et identique pour toutes les stations.

### 3 Un peu de stats (5 points)

Le département de contrôle qualité d'une entreprise a mis en œuvre un système de détection de défauts dans la production de ses produits. Le procédé consiste à effectuer  $N$  (constante égale à 30) mesures du diamètre d'une pièce, calculer sa valeur moyenne et sa variance. Les données traitées dans ce programme sont organisées dans une structure `mesuresUnePiece` avec les champs : un tableau pouvant contenir  $N$  mesures du diamètre d'une pièce, le nombre de mesures prises, la moyenne et la variance des valeurs mesurées (calculées après la phase de mesure).

3.1 Ecrire une fonction `int mesure(...)`, qui prend en entrée la structure de données et permette à un ouvrier d'enregistrer par le clavier plusieurs mesures du diamètre d'une pièce à la fin de la ligne de production. La fonction retournera le nombre de mesures effectuées et les mesures.

3.2 Ecrire une fonction récursive `float sum(...)` qui prend en entrée un tableau de mesures et son nombre d'éléments, et retourne la somme de ces éléments.

3.3 Ecrire une fonction `void calculMoyVar(...)` qui prend en entrée la structure de données. Cette fonction calculera la moyenne et la variance des mesures (les fonctions des questions 1 et 2 seront utilisées). On prendra comme formule pour la variance : la moyenne des carrés des écarts à la moyenne.

### 4 Capteurs (5 points)

Soit une chaîne de caractère, issue d'un capteur de mesures, comportant les éléments formatés suivants :

"nom-capteur nombre-mesures-n mesure-1 mesure-2 mesure-3 ... mesure-n"

Exemple : "capteur-1 5 -2 5 9 -15 10" avec

- "capteur-1" représentant le nom du capteur, **ce nom fait toujours exactement 9 caractères**,
- "5" représentant le nombre 'n' (strictement positif) de mesures numériques effectuées par le "capteur-1",
- "-2", "5", "9", "-15" et "10" représentant les valeurs des 'n=5' mesures numériques effectuées par le "capteur-1".

4.1 Programmez la fonction `int* transcodeTabChar(char* tab)` qui transcodera un tableau de caractères `tab` (contenant une chaîne de caractères au format expliqué précédemment) en un **tableau d'entiers** relatifs au format suivant :

- le 1<sup>er</sup> élément du tableau d'entiers **SERA** le nombre 'n' de mesures numériques effectuées,
- les autres éléments du tableau d'entiers seront les valeurs des 'n' mesures numériques effectuées.

Exemple pour la chaîne "capteur-1 5 -2 5 9 -15 10" : le tableau d'entiers contiendra '{5,-2,5,9,-15,10}'

Le tableau d'entiers sera alloué dynamiquement dans la fonction `transcodeTabChar()`.

4.2 Soit 2 tableaux d'entiers relatifs 'tab1' et 'tab2' différents (nombre d'éléments et éléments différents), obtenus par 2 appels à la fonction précédente `transcodeTabChar()`.

Programmez la fonction `int* multiplexTabInt(int* tab1,int* tab2)` qui effectuera un multiplexage des tableaux 'tab1' et 'tab2' qu'elle retournera dans un tableau d'entiers relatifs.

Multiplexage signifie que la fonction prendra l'élément 1 de 'tab1' et le stockera en position 1 du tableau multiplexé d'entiers. Puis elle prendra l'élément 1 de 'tab2' et le stockera en position 2 du tableau multiplexé d'entiers. Puis elle prendra l'élément 2 de 'tab1' et le stockera en position 3 du tableau multiplexé d'entiers. Et ainsi de suite jusqu'au stockage, dans le tableau multiplexé d'entiers, de tous les éléments de 'tab1' et de 'tab2'.

Exemple pour 'tab1={5,-2,5,9,-15,10}' et 'tab2={3,9,-45,28}' : le tableau multiplexé d'entiers contiendra '{5,3,-2,9,5,-45,9,28,-15,10}'.

Le tableau multiplexé d'entiers sera alloué dynamiquement dans la fonction `multiplexTabInt()`.