

Numéro d'Anonymat :

LIF 1 : Algorithmique et Programmation Impérative, Initiation

Durée : 2h

Contrôle Continu Terminal : Mercredi 14 Janvier 2009

Les documents, calculatrice, téléphone portable sont interdits

La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

Vous veillerez à respecter les notations et les règles d'écriture des algorithmes vues en cours et en TD. Soyez attentifs aux indices des tableaux et prenez garde à ne pas dépasser la fin des tableaux. Le barème est donné à titre indicatif.

Partie A : Questions de cours (5 pts)

```
#include <iostream>
#include <string>
using namespace std;
const int MAX_CHAINE 50;
void affiche_resultat(int tab[26])
{
    int i;
    char c;
    for (i=0;i<26;i++);
    {
        c=i+"a";
        cout<< c << "|" << tab[i] << endl;
    }
}
void mystere(char chaine[MAX_CHAINE], int &tab[26])
{
    int i,lg, indice;
    lg= strlen(ch);

    for (i=0,i<lg,i++)
    {
        indice == chaine[i]-'a';
        tab[indice]++
    }
}
int main (void)
{
    char ch[MAX_CHAINE];
    int tab[26]={0};
    cin<<ch;
    mystere(tab,ch);
    affiche_resultat(tab);
    system("PAUSE");
    return 0;
}
```

- 1- Corrigez les erreurs syntaxiques directement sur le programme ; vous rendrez le sujet avec votre copie en recopiant le numéro d'anonymat en haut de la page.(3 pts)
- 2- Expliquez ce que fait le programme ci-dessus. (2 pts)

Partie B : Algorithmique (7 pts)

- 1- Écrire l'algorithme d'un sous-programme qui retourne VRAI si le tableau T de 11 entiers passé en paramètres contient plus de valeurs paires que de valeurs impaires et FAUX sinon.
- 2- Écrire l'algorithme d'une fonction qui prend en argument deux chaînes de caractères et retourne la position de la première occurrence de la chaîne2 dans la chaîne1 si elle est présente et -1 sinon.

Exemples : recherche_motif ("bonneannee", "an") → 5
recherche_motif ("bonnesante", "voeux") → -1

- 3- On souhaite comptabiliser le nombre d'occurrences de chacun des caractères contenus dans une chaîne de caractères **composée exclusivement des caractères alphabétiques minuscules**. Ecrivez l'algorithme d'un sous-programme permettant de remplir un tableau dont chacune des 26 cases contiendra le nombre d'apparitions d'un caractère particulier dans la chaîne.

Exemple : Soit la chaîne "voiciunechainecomposeeuniquementdeminuscules"

nombre de a	nombre de b	nombre de c	nombre de d	nombre de e	nombre de f	nombre de g	nombre de h	nombre de i	nombre de j	nombre de k	nombre de l	nombre de m	nombre de n	nombre de o	nombre de p	nombre de q	nombre de r	nombre de s	nombre de t	nombre de u	nombre de v	nombre de w	nombre de x	nombre de y	nombre de z
1	0	4	1	8	0	0	1	5	0	0	1	3	5	3	0	1	0	3	1	5	1	0	0	0	0

Partie C : Langage C (8 pts)

1. Définir en langage C un type *complexe* pour représenter les nombres complexes.
2. Écrire une fonction qui prend en paramètre deux flottants, et qui retourne un nombre complexe. (Remplissage de la structure)
3. Écrire une fonction qui affiche un nombre complexe (sous la forme $a+ib$)
4. Définir en langage C une structure permettant de stocker un tableau de 10 nombres complexes et mémorisant de plus le nombre de complexes déjà présents dans le tableau.
5. Écrire en langage C un sous-programme permettant de remplir le tableau précédent avec un nombre complexe saisi par l'intermédiaire de la fonction écrite en 2.
6. Écrire en langage C un sous-programme permettant d'afficher le tableau. On fera attention de n'afficher que les cases comportant un nombre complexe (et non pas les cases "vides").
7. Écrire la fonction somme_complexe permettant de réaliser l'addition des n nombres complexes contenus dans le tableau précédent ; n étant le nombre de cases remplies dans le tableau. On rappelle que la somme de 2 nombres complexes peut s'écrire de la manière suivante : $(a_1+ib_1)+(a_2+ib_2) = (a_1+a_2)+i(b_1+b_2)$
8. Écrire le programme principal proposant un menu disposant des choix suivants à l'utilisateur (1- Quitter le programme 2- Saisir une nombre complexe 3- Afficher un nombre complexe 4- Faire la somme de deux nombres complexes) et permettant d'appeler les sous-programmes précédemment écrits.