


Questionnaire Contrôle Périodique

INF1005C

Sigle du cours

| Identification de l'étudiant(e) | | |
|---------------------------------|-------------|----------|
| Nom : | Prénom : | |
| Signature : | Matricule : | Groupe : |

| Sigle et titre du cours | | Groupe | Trimestre |
|---|-----------------|---|-----------|
| INF1005C – Programmation procédurale | | 1005 | 20091 |
| Professeur | | Local | Téléphone |
| Martine Bellaïche, responsable – Jean-Charles Bernard, professeur | | M 414 | 4679 |
| Jour | Date | Durée | Heures |
| Vendredi | 27 février 2009 | 1h55 | 8h30 |
| Documentation | | Calculatrice | |
| <input type="checkbox"/> Toute <input checked="" type="checkbox"/> Aucune <input checked="" type="checkbox"/> Voir directives particulières | | <input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Programmable <input type="checkbox"/> Non programmable Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits. | |
| Directives particulières | | | |
| <p> Vous n'avez pas à écrire de commentaires ni d'en-têtes.</p> | | | |

Bonne chance à tous!

Important

Cet examen contient **4** questions sur un total de **4** pages (excluant cette page)

La pondération de cet examen est de **25** %

Vous devez répondre sur : ☐ le questionnaire ☒ le cahier ☐ les deux

Vous devez remettre le questionnaire : ☐ oui ☒ non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

Question 1 (6 points)

- 1.1 Soient les déclarations suivantes :

```
int compteur = 0, limite = 20;
```

Déterminez si les expressions suivantes sont vraies ou fausses.

```
(compteur == 0) && (limite < 20)
```

```
(limite > 20) || (compteur < 5)
```

```
!(compteur == 12)
```

- 1.2 Dans la boucle ci-dessous, donnez combien de fois la condition $n > 0$ est exécutée.

```
int n ;  
for ( n = 10; n > 0; n = n-2)  
{  
    cout << "Bonjour " << n << endl;  
}
```

- 1.3 Afin d'obtenir le même affichage, modifiez le programme ci-dessous en utilisant l'instruction de répétition `for` au lieu de l'instruction de répétition `while`.

```
int nombre;  
cout << "Veuillez rentrer le nombre de fois";  
cin >> nombre;  
i = 1;  
while ( i <= nombre)  
{  
    if (i < 5 && i!=2)  
        cout << 'X';  
    i++;  
}
```

- 1.4 Afin d'obtenir le même affichage, modifiez le programme ci-dessous en utilisant l'instruction de répétition `for` au lieu de l'instruction de répétition `do while`.

```
int nombre;
cout << "Veuillez rentrer le nombre de fois ";
cin >> nombre;
i = 100;
do
{
    cout << 'X';
    i = i+100;
} while ( i < nombre);
```

Réponse Question 1

1.1 faux

vraie

vraie

1.2 6 fois

1.3

```
int nombre;
cout << "Veuillez rentrer le nombre de fois";
cin >> nombre;
for ( i = 1 ; i <= nombre; i++)
    if (i < 5 && i!=2)
        cout << 'X';
```

1.4

```
int nombre;
cout << "Veuillez rentrer le nombre de fois";
cin >> nombre;
cout << 'X';
for ( i = 200 ; i < nombre; i= i+100)
    cout << 'X';
cout << endl;
```

Question 2 (3 points)

Deux mots sont des anagrammes s'ils ont les mêmes lettres, mais forment un mot différent. Par exemple chien et niche. Le début du programme ci-dessous demande à l'utilisateur deux mots et calcule les fréquences des lettres de chacun des mots. On vous demande de compléter le programme qui permet de vérifier si les deux mots sont des anagrammes et d'afficher le résultat de la vérification. Vous pouvez déclarer toutes les variables nécessaires à la poursuite du programme.

```
// initialiser les tableaux à zero
int frequenceMot1[26]={0}, frequenceMot2[26]={0};

string mot1, mot2;

cout << "entrer le premier mot ";
cin>> mot1;
cout << "entrer le deuxieme mot ";
cin >> mot2;

// calcul des fréquences des lettres
// de chaque mot
for ( i = 0; i < mot1.size(); i++)
    frequenceMot1[mot1[i]-'a']++;

for ( i = 0 ; i < mot2.size(); i++)
    frequenceMot2[mot2[i] -'a']++;

// à completer
```

Réponse Question 2

```
bool estAnagramme = true;
for (i = 0; (i < 26) && (estAnagramme); i++)
{ if (frequenceMot1[i] != frequenceMot2[i])
    estAnagramme = false;
}
if (estAnagramme)
    cout << mot1 << "\t" << mot2 << " sont des anagrammes "<< endl;
else
    cout << mot1 << "\t" << mot2 << " ne sont pas des anagrammes "<< endl;
```

Question 3 (5 points)

Sur chaque ligne d'un fichier texte, on trouve dans l'ordre, une chaîne de caractères pouvant contenir des espaces, une tabulation (<tab>), le nombre d'espaces que contient la chaîne et une fin-de-ligne (<enter>). Par exemple,

```
Dimanche au soir a Chateaugay <tab> 4 <enter>
Les pieds pendants au bout du quai (ter)<tab>7<enter>
La riviere joue de l'harmonica<tab>4 <enter>
Ma femme se baigne les pieds dans l'eau<tab>7<enter>
. . . . .
. . . . .
<eof>
```

Écrivez le programme qui réalise les tâches suivantes :

- le nom du fichier texte est fourni par l'utilisateur suite à une invite du programme,
- le fichier est ouvert et après lecture de chaque ligne, les informations lues sont affichées dans le format suivant :
la chaîne de caractères Dimanche au soir a Chateaugay
contient 4 espaces
- la somme totale des espaces des lignes est calculée et affichée.

Notes :

Le nombre de lignes du fichier est inconnu.

Vérifiez si le fichier existe et s'il n'est pas vide.

On peut lire une phrase jusqu'au caractère de tabulation <tab> à l'aide de l'instruction suivante.

```
ifstream lire;
string ligne;
getline(lire, ligne, '\t');
```

Réponse question 3

```
void main()
{
    ifstream lire;
    string nomFichier, ligne;
    int totLignes, nbBlancs, totNbBlancs=0;
    cout<<" Nom du fichier ";
    cin>>nomFichier;
    totLignes=0;
    lire.open(nomFichier.c_str());
    if (lire.fail()) cout<<"\n\n Erreur d'ouverture du fichier!\nArret du
programme";
    else
    {
        cout<<endl<<endl;
        getline(lire, ligne, '\\t');

        while (!lire.eof())
        {
            totLignes++;
            lire>>nbBlancs;
            cout<<ligne<<" contient "<<nbBlancs<<" espaces"<<endl;
            totNbBlancs += nbBlancs;
            lire.ignore(1);
            getline(lire, ligne, '\\t');
        } // fin du while
        cout<<"\n\nil y a un total de "<<totNbBlancs<<" espaces dans le
texte";
        cout<<"\n\n"<<totLignes<<" lignes ont ete lues";
        lire.close();
    } // fin du else
}
```

Question 4 (6 points)

Lorsque l'on tape un mot au clavier, l'utilisateur effectue fréquemment une inversion des deux lettres adjacentes du mot.

Soit la déclaration suivante :

```
string mot = "programmation";
```

Le programme demande à l'utilisateur un mot à vérifier. Ensuite, tant que le programme n'a pas trouvé, que le mot à vérifier est égal à la variable mot, il doit inverser toutes les lettres adjacentes du mot à vérifier pour le comparer à la variable mot.

Par exemple, si le mot à vérifier est `prgorammation`, le programme vérifie avec les inversions suivantes : `rpgorammation`, `pgrorammation`, `programmation`. Les permutations sont alors terminées, car la dernière permutation a donné le bon mot. On affiche alors la position de l'erreur à partir de la valeur 0. Dans notre exemple, on affiche la valeur 2.

Écrivez le programme permettant de résoudre ce problème.

Réponse question 4

```
int i , j;
string mot = "programmation";

string motAVerifier, motTemporaire;
char permut;
bool trouver = false;
cout << "Veuillez rentrer le mot a verifier ";
cin >> motAVerifier;

if (mot == motAVerifier)
    cout << "Pas de faute frappe"<< endl;
else if (mot.size() == motAVerifier.size())
{
    for ( i = 1 ; (i < mot.size()) && (!trouver); i++)
    {
        motTemporaire = mot;
        permut = motTemporaire[i-1];
        motTemporaire[i-1] = motTemporaire[i];
        motTemporaire[i] = permut;
        if (motTemporaire == motAVerifier)
        {
            trouver = true;
            cout << "l'erreur se trouve a la position " << i-1 <<
endl;
        }
    }
}
else
    cout << "Le mot a une taille différente " << endl;
```