

Questionnaire Intra +corrigé

INF1005C

Sigle du cours

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

Sigle et titre du cours		Groupe	Trimestre
INF1005C – Programmation procédurale		Tous	20121
Professeur		Local	Téléphone
Martine Bellaïche, responsable et Jean-Charles Bernard		M-3414	4679
Jour	Date	Durée	Heures
Vendredi	2 mars 2012	1h50	8h30-10h20

Documentation	Calculatrice	
<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toute <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
<ul style="list-style-type: none"> Ne recopiez pas les déclarations, ni les instructions déjà fournies dans le questionnaire. Vous n'avez pas à écrire de commentaires, ni d'en-têtes, ni les includes. On ne répondra à aucune question. En cas de doute, veuillez faire vos suppositions et les écrire sur le cahier d'examen.

Important
Cet examen contient 4 questions sur un total de 3 pages (excluant cette page)
La pondération de cet examen est de 25 %
Vous devez répondre sur : <input type="checkbox"/> le questionnaire <input checked="" type="checkbox"/> le cahier <input type="checkbox"/> les deux
Vous devez remettre le questionnaire : <input type="checkbox"/> oui <input checked="" type="checkbox"/> non

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

Question 1 (5 points)

Le programme suivant demande à l'utilisateur un nombre de fois, une borne inférieure et supérieure, génère des nombres aléatoires et compte le nombre de nombres aléatoires entre les bornes inférieure et supérieure. Le programme comporte des erreurs de compilation et d'exécution. Identifier les lignes comportant des erreurs et les corriger.

```
1. int main()
2. {
3.     int inferieur,superieur, nombrefois;
4.     int nombreAleatoire, nombreFoisDansIntervalle;
5.     float pourcentageEntreInfSup;
6.     srand(time(0));

7.     cout << "donner la borne inferieure ";
8.     cin << inferieur;
9.     cout << "Donner la borne superieure ";
10.    cin << superieur;
11.    cout << "donner le nombre de fois ";
12.    cin << nombreFois;

13.    if (inferieur < superieur || nombreFois >= 0)
14.    { for (i = 0; i < nombreFois; i++);
15.        nombreAleatoire = rand();
16.        if((nombreAleatoire > inferieur) && (nombreAleatoire < superieur))
17.            nombreFoisDansIntervalle++;
18.        pourcentageEntreInfSup = nombreFoisDansIntervalle/nombreFois*100;
19.        cout << "Le pourcentage de nombres entre "<< inferieur
20.            << " et " << superieur<< " est " <<pourcentageEntreInfSup<<endl;
21.    }
22.    else
23.        cout << inferieur << " n'est pas plus petit que"<< superieur<<endl;
24.        cout << " ou le nombre de fois n'est pas  > = 0" <<endl;
25.
26. }
```

Question 2 (5 points)

Écrire un programme qui demande à l'utilisateur un mot (sans espace) et détermine si le mot commence par une voyelle ou une consonne. Jusqu'à la fin du cin (`cin.eof()` correspond au ctrl+z au clavier), le programme répète l'entrée du mot par l'utilisateur. Si le mot commence par une voyelle, il est stocké dans le tableau `listeMotsVoyelle`, et si le mot commence par une consonne, il est stocké dans le tableau `listeMotsConsonne`. Les variables `nombreVoyelles` et `nombreConsonnes` permettent de connaître le nombre de mots commençant par une voyelle et par une consonne. Par la suite, afficher le contenu des variables `nombreVoyelles`, `listeMotsVoyelle`, `nombreConsonnes` et `listeMotsConsonne`. Le mot peut commencer par une majuscule ou une minuscule. Il n'est pas nécessaire de vérifier le débordement des tableaux.

```
const int MAX = 100;
string listeMotsVoyelle[MAX];
int nombreVoyelles = 0;
string listeMotsConsonne[MAX];
int nombreConsonnes = 0;
string mot;
```

Notes :

<i>tolower(int argument)</i>	Convertit un caractère en minuscule.
<i>toupper(int argument)</i>	Convertit un caractère en majuscule.

Question 3 (4 points)

Soient les déclarations suivantes :

```
const int MAX = 8;  
int ordre [MAX] = {55,27,30,14,59,65,20,81};  
int desordre[MAX];
```

En utilisant les déclarations précédentes et en déclarant toutes les déclarations nécessaires, écrire les instructions qui placent les valeurs du tableau `ordre`, dans le tableau `desordre` dont les positions des valeurs dans ce tableau `desordre` sont choisies aléatoirement. En d'autres mots :

La première valeur 55 étant à la position 0 du tableau `ordre`, si on obtient la valeur aléatoire 3 entre [0, MAX], la valeur 55 sera placée à la position 3 du tableau `desordre`.

La deuxième valeur 27 étant à la position 1 du tableau `ordre`, on trouve une valeur aléatoire différente de 3 et entre [0, MAX], soit 1, la valeur 27 sera placée à la position 1 du tableau `desordre`.

La troisième valeur 30 étant à la position 2 du tableau `ordre`, on trouve une valeur aléatoire différente de 3 et 1 et entre [0, MAX], soit 5, la valeur 30 sera placée à la position 5 du tableau `desordre`.

Et ainsi de suite.

Notes :

1. Pour générer des nombres aléatoires, on fait appel une seule fois à la fonction `srand(time(0))`;
2. Ensuite, on peut faire appel plusieurs fois à la fonction `rand()` pour générer un nombre aléatoire entre [0, `LePlusGrandEntier`].

Question 4 (6 points)

On vous demande d'aider Hydro-Québec à établir les factures de leurs clients. Hydro-Québec effectue un relevé du compteur d'électricité à peu près tous les deux mois. Donc par client, Hydro-Québec a un fichier de consommation ayant le nom du client et 3 valeurs par deux mois. La première valeur est le nombre de jours, la deuxième valeur est la consommation en KWatt des 30 premiers jours et la troisième valeur en KWatt est la consommation des jours suivants. Le fichier de consommation a le format suivant pour le client temblay:

```
58      ←----- nombre de jours
1740    ←----- consommation les 30 premiers jours
4920    ←----- consommation les jours suivants
61      ←----- nombre de jours
1830    ←----- consommation les 30 premiers jours
1100    ←----- consommation les jours suivants
64
1470
0
57
1060
.....
```

Hydro Québec établit les coûts suivants :

- la redevance d'abonnement qui a le coût : nombre de jours * 0.4064 \$
- les 30 premiers jours qui a le coût : consommation 30 premiers jours * 0.0545\$
- les jours suivants qui a le coût : consommation des jours suivants * 0.0746\$

Pour le nombre de jours, le programme devra calculer les coûts de la redevance d'abonnement, la somme en KWatt de la consommation des 30 premiers jours et des jours suivants, le coût des 30 premiers jours et le coût des jours suivants. Ensuite le coût total pour la somme de tous les jours.

Écrire un programme qui

- demande à l'utilisateur le nom du client;
- ouvre le fichier de consommation ayant comme nom, le nom du client suivi de l'extension .txt;
- crée le fichier ayant comme nom, le nom du client concaténé de « facture.txt » ;
- lit tout le fichier dont on ne connaît pas le nombre de lignes;
- sauvegarde dans un fichier de facture, le nom du client, et pour chaque nombre de jours :
 - ✓ la consommation totale des KWatt,
 - ✓ le coût de la redevance d'abonnement,
 - ✓ le coût des 30 premiers jours
 - ✓ le coût des jours suivants
- et le coût total pour l'ensemble des jours.

Question 4 (suite)

Notes :

- Vérifier l'existence du fichier.
- Vérifier si le fichier n'est pas vide.
- Vérifier la création du fichier.

Par exemple, le fichier de sortie pour le client tremblay aura le contenu suivant :

```
tremblay
Redevance d'abonnement 23.5712
Consommation Total 6660
30 premiers jours 94.83
reste de consommation 367.032
Redevance d'abonnement 24.7904
Consommation Total 2930
30 premiers jours 99.735
reste de consommation 82.06
Redevance d'abonnement 26.0096
Consommation Total 1470
30 premiers jours 80.115
reste de consommation 0
Montant Global 798.143
```

Réponse question 1 (5 points)

```
int main()
{
    int inferieur, superieur, nombreFois, i ; 0.5 point
    int nombreAleatoire, nombreFoisDansIntervalle = 0; 1 point
    float pourcentageEntreInfSup;
    srand(time(0));

    cout << "donner la borne inferieure ";
    cin >> inferieur; 0.5 point
    cout << "Donner la borne superieure ";
    cin >> superieur;
    cout << "donner le nombre de fois ";
    cin >> nombreFois;

    if (inferieur < superieur && nombreFois >= 0) 0.5 point
    { for (i = 0; i < nombreFois; i++) // retirer ; 0.5 point
        { 0.5 point

            nombreAleatoire = rand();
            if((nombreAleatoire > inferieur) && (nombreAleatoire < superieur))
                nombreFoisDansIntervalle++;

        }

        pourcentageEntreInfSup =
float (nombreFoisDansIntervalle)/(float) nombreFois*100; 0.5 point
        cout << "Le pourcentage de nombres entre "<< inferieur
            << " et " << superieur<< " est "
<<pourcentageEntreInfSup<<endl;
    }
    else 0.5 point
    {
        cout << inferieur << " n'est pas plus petit que"<< superieur<<endl;
        cout << " ou le nombre de fois n'est pas > = 0" <<endl;
    }
    return 0; 0.5 point
}
```

Réponse question 2 (5 points)

```

int main()
{
    const int MAX = 100;
    string listeMotsVoyelle[MAX];
    int nombreVoyelles = 0;
    string listeMotsConsonne[MAX];
    int nombreConsonnes = 0;
    string mot;

    lecture 0.5 point

    cout << " veuillez donner un mot ";
    cin >> mot;
    while (!cin.eof())    boucle 1.0
    {
        Comparaison voyelle et consonne 1.0 point
        affectation et comptage 1.0 point

        if (mot[0] == 'a' || mot[0] == 'e' ||
            mot[0] == 'i' || mot[0] == 'o' ||
            mot[0] == 'u' || mot[0] == 'y')

        {
            listeMotsVoyelle[nombreVoyelles] = mot;
            nombreVoyelles++;
        }
        else
        {
            listeMotsConsonne[nombreConsonnes] = mot;
            nombreConsonnes++;
        }

        relecture 0.5 point

        cout << " veuillez donner un mot ";
        cin >> mot;
    }

    Affichage 0.5 point
    Boucle 0.5 point

    cout << " le nombre de mots commençant par une voyelle "
        << nombreVoyelles << endl;
    for (int i = 0 ; i < nombreVoyelles; i++)
        cout << listeMotsVoyelle[i] << endl;
    cout << endl;
    cout << " le nombre de mots commençant par une consonne "
        << nombreConsonnes << endl;
    for (int i = 0 ; i < nombreConsonnes; i++)
        cout << listeMotsConsonne[i] << endl;
    cout << endl;

    return 0;
}

```


Réponse question 3 (4 points)

```
int main()
{
    //déclarations 0.5 points
    const int MAX = 8;
    int ordre [MAX] = {55,27,30,14,59,65,20,81};
    int indiceAleatoire[MAX];
    int desordre[MAX];
    int i , j, alea;
    bool trouve;

    //valeurs aleatoires 1.0 point
    srand(time(0));

    indiceAleatoire[0] = rand()% MAX;

    //parcours de tout le tableau 0.5 point
    for ( i = 1 ; i < MAX; i++)
    {
        //Placer dans un ordre aleatoire 1.5 points
        do
        {
            trouve = false;
            alea = rand() % MAX;
            for (j =0 ; j < i ; j++)
                if (alea == indiceAleatoire[j])
                    trouve = true;

        } while (trouve);
        indiceAleatoire [i] = alea;
    }

    //Placer en désordre 0.5 points
    for ( i = 0 ;i < MAX; i++)
        desordre[i] = ordre[indiceAleatoire[i]];

    return 0;
}
```

Réponse question 4 (6 points)

```
int main()
{
    declarations 0.5 points
        ifstream ficConsommation;
        ofstream ficFacture;
        string nomFichierConsommation;
        string nomFichierFacture;

        int nombreJours;
        float consommation30, consommation30Plus ;
        float totalConsommation, total, montantGlobal = 0;
    nom du fichier 0.5 points
        cout << "donner le nom de l'utilisateur " ;
        cin >> nomFichierConsommation;
    ouverture des fichiers 0.5 points
        ficConsommation.open (nomFichierConsommation+".txt");
        ficFacture.open (nomFichierConsommation+"Facture.txt") ;
    vérification 0.5 points
        if (! ficConsommation.fail() || ficFacture.fail())
        {
            premiere lecture , non fin de fichier, relecture 1.5 points
                ficFacture << nomFichierConsommation<<endl;
                ficConsommation>> nombreJours;
                ficConsommation >> consommation30;
                ficConsommation >> consommation30Plus;

            while (!ficConsommation.eof())
            {
                calcul des coûts 1.0 point
                Ecriture dans le fichier 0.5 point
                    total = nombreJours*0.4064;
                    montantGlobal += total;
                    ficFacture << "Redevance d'abonnement " << total <<endl;

                    ficFacture << "Consommation Total "
                        << consommation30 + consommation30Plus<<endl;

                    total = consommation30 * 0.0545;
                    montantGlobal +=total;
                    ficFacture << "30 premiers jours " <<total <<endl;

                    total = consommation30Plus *0.0746;
                    montantGlobal += total;
                    ficFacture << "reste de consommation " << total <<endl;

                    ficConsommation>> nombreJours;
                    ficConsommation >> consommation30;
                    ficConsommation >> consommation30Plus;
            }
            Calcul nontant global 0.5 points
                ficFacture << "Montant Global " << montantGlobal<<endl;
            fermeture des fichiers 0.5 points
                ficConsommation.close();
                ficFacture.close();
        }
    else
        cout << " impossible d'ouvrir ou de créer les fichiers"<<endl;
```

```
return 0 ;  
}
```