ÉCOLE POLYTECHNIQUE MONTRÉAL

Le génie sans frontières

Questionnaire
Contrôle Périodique
+ 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**	**20091**
Professeur	***Local***	***Téléphone***
Martine Bellaïche, responsable – Jean-Charles Bernard, professeur	**M-3414**	**4679**

Jour	*Date*	*Durée*	*Heures*
Vendredi	**26 février 2010**	**1h50**	**8h30**

Documentation	*Calculatrice*	
☐ Toute ☒ Aucune ☒ Voir directives particulières	☒ Aucune ☐ Programmable ☐ Non programmable	**Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.**

Directives particulières

Vous n'avez pas à écrire de commentaires ni d'en-têtes., ni les include au début du programme.

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 (5 points)

1.1 Quel sera l'affichage du programme suivant s'il est exécuté ?

```
const int MAX = 3;
int Matrice[MAX][MAX], Ligne, Colonne;
for (Ligne=0; Ligne<3 ; Ligne++)
    for (Colonne=0; Colonne<MAX; Colonne++)
         Matrice[Ligne][Colonne]=0 ;
 for (Ligne=0; Ligne<MAX; Ligne++)
    for (Colonne=Ligne; Colonne>=0; Colonne--)
          Matrice[Ligne][Colonne]=Ligne;
 for (Ligne=0; Ligne<MAX ; Ligne++)
 {
     for (Colonne=0; Colonne<MAX; Colonne++)
         cout<<Matrice[Ligne][Colonne] << " ";
     cout << endl;

 }
```

1.2 Quel sera l'affichage des instructions suivantes si elles sont exécutées ?

```
 int A = 5, B = 2;
 cout << A/B<<endl;
```

1.3 Quel sera l'affichage des instructions suivantes si elles sont exécutées ?

```
int Mois;
float Temperature = 0 , Humidite =0 ;
for (Mois = 1 ; Mois < 3 ; Mois++)
       if (Mois >= 6)
           Temperature = 30.4;
           Humidite = 90.23;
           cout << "Temperature " << Temperature << endl;
           cout << "Humidite " << Humidite<<endl;
```

1.4 Donner les déclarations des variables permettant d'exécuter les instructions ci-dessous.

```
 nbLettres= mot.size();
 if (reponse == 'N')
        reste = nombre%2;
 if (!trouve)
         cout <<"pas de chance"<<endl;

getline(cin,phrase);
```

Réponse question 1

1.1

0 0 0

1 1 0

2 2 2

1.2

2

1.3

Temperature 0

Humidite 90.23

1.4

```
int  nbLettres ;
string mot;
char  reponse;
int  nombre, reste;
bool trouve;
string phrase;
```

Question 2 (5 points)

Une grille G de dimension 20 x 20 est représentée par un tableau à 2 dimensions de valeurs booléennes. Au départ, un pion est placé sur cette grille. Les déplacements du pion sur la grille sont choisis aléatoirement. Le pion est placé dans sa nouvelle position que si celle ci est dans les limites de la grille. En cas de débordement des limites, on ne déplace pas le pion. La position X du pion est représentée par le nombre de lignes du tableau et la position Y du pion est représentée par les colonnes du tableau. Le pion dans la grille, sera identifié par la valeur true à une position dans le tableau.

Écrire un programme qui permet de simuler le déplacement d'un pion dans la grille. Le programme devra réaliser les opérations suivantes.

- Initialiser le tableau à la valeur false.
- Demander à l'usager le nombre de déplacements que le pion doit effectuer.
- Déterminer la position initiale (X,Y) du pion. Cette position est une valeur aléatoire entre les valeurs [0,19]. Selon la position déterminée (X,Y), mettre la valeur du tableau à true pour placer le pion.
- Tant que le nombre de déplacements du pion n'est pas atteint
    - Calculer 2 nombres aléatoires entre [-5, 5]. Ces deux nombres représentent les déplacements en X et Y.
    - Vérifier si en ajoutant le déplacement en X et Y à la position actuelle du pion, on ne déborde pas du tableau.
    - Afficher un message d'erreur pour le débordement en Y.
    - Afficher un message d'erreur pour le débordement en X.
    - Si non débordement du tableau,
        - Déplacer le pion à la nouvelle position correspondant à la somme des déplacements et de l'ancienne position.
        - Afficher la position actuelle en X et Y du pion.
        - Augmenter le nombre de déplacements.

Notes :

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

Question 2

```
int main()
{
      const int DIM = 20;
      bool  grille[DIM][DIM] ;
      int i ,j ;
      int deplacementX, deplacementY;
      int nbDeplacements;
      int positionX, positionY;
      bool deplaceX, deplaceY;
      srand(time(0));
       for (  i = 0; i < DIM; i++)
             for (  j = 0; j < DIM; j++)
                   grille [i][j] = false;
       cout << "Combien de deplacements voulez vous effectuer ";
       cin >>  nbDeplacements;
       positionX = rand()%20;
       positionY = rand()%20;
       cout << "position initiale " << positionX << " "<< positionY <<endl;
       grille[positionX][positionY] = true;
      i = 0;
       while ( i < nbDeplacements)
       {  deplacementX = rand()%11-5;
          deplacementY = rand()%11-5;
           deplaceX = deplaceY = false;
           if ((positionX+deplacementX >= 0) && (positionX+deplacementX <20))
                 deplaceX = true;

           else
                 cout <<"debordement de  la  grille en X"<<endl;

        if ((positionY+deplacementY >= 0) && (positionY+deplacementY <20))
              deplaceY = true;

         else
                 cout <<"debordement de  la  grille en Y"<<endl;
        if (deplaceX && deplaceY )
         {   grille[positionX][positionY] = false;
             positionX+=deplacementX;
             positionY += deplacementY;
             cout << "nouvelle position du pion " << positionX
                  << " "<<positionY<<endl;
             grille[positionX][positionY] = true;
             i++;
         }
       }

       return 0;
}
```

Question 3 (5 points)

Écrire un programme jouant au jeu du pendu. L'usager qui interagit avec le programme, doit deviner un mot lu d'un fichier. L'usager donne une lettre (minuscule ou majuscule) au programme, mais pas de chiffre. Le programme vérifie si cette lettre appartient au mot et place la lettre à la bonne position. Et ainsi de suite jusque l'usager trouve toutes les lettres du mot à deviner. Au début du jeu, on affiche une série X correspondant au nombre de lettres à deviner. Faire les affichages appropriés. Les instructions correspondantes à la lecture du mot à deviner dans le fichier, ne sont pas à faire. Déclarer toutes les variables utiles à la réalisation du jeu.

Notes :

- On suppose que le mot à deviner ne contient pas de lettre X.
- La fonction `tolower('A')` a la valeur `'a'` et `tolower('a')` a la valeur `'a'`.

```
int main()
{
  string motDevine; // ce mot est lu dans un fichier.
 // NE PAS FAIRE : lecture du mot dans le fichier
.....
}
```

Voici un exemple d'affichage du programme.

XXXXXX
Donner une lettre h
XXXXXX
Donner une lettre I
XXXXXX
Donner une lettre T
tXXtXX
Donner une lettre o
toXtXX
Donner une lettre e
toXtXe
Donner une lettre r
tortXe
Donner une lettre a
tortXe
Donner une lettre u
Bravo vous avez trouve
tortue

Question 3

```
int main()
{
  string motDevine ;
  // ce mot est lu dans un fichier.
 // NE PAS FAIRE : lecture du mot dans le fichier

  string motTrouve;
  char lettre;
  int i;
  bool trouve = false;

  for ( i = 0; i < motDevine.size() ; i++)
          motTrouve+='X';

  while (!trouve)
  {   cout << motTrouve<<endl;
      cout << "Donner une lettre ";
      cin  >> lettre;
      for ( i = 0 ; i < motDevine.size() ; i++)
           if (motDevine[i] == tolower(lettre))
                 motTrouve[i] = tolower(lettre);

      if (motTrouve == motDevine)
      {   trouve = true;
          cout << "Bravo vous avez trouve" <<endl;
        cout << motTrouve<<endl;
      }
  }
  return 0;

}
```

Question 4 (5 points)

Chaque semaine, une entreprise de vente au détail offre à ses clients un ensemble d'items en promotion . À la fin de chaque semaine, le résultat des ventes des produits en promotion est analysé afin de déterminer l'ensemble des produits en promotion pour la semaine suivante. L'entreprise procède comme suit :

> Pour chacun des items, elle compare la quantité en inventaire au début de la semaine avec le nombre d'items vendus.
> Si le rapport entre le nombre vendu et le nombre en inventaire dépasse 37%, l'item est maintenu pour la promotion de la semaine suivante et une demande d'ajustement d'inventaire est adressée à son fournisseur.
> Si ce rapport se situe entre 17% et 37%, l'item est maintenu pour la promotion de la semaine suivante sans demande d'ajustement d'inventaire.
> Si la valeur du rapport est inférieure à 17%, l'item est retiré de la liste des items en promotion.

Écrire un programme qui lit le contenu du fichier texte ventes.txt, l'analyse et affiche le nom des items nécessitant un avis de mise à jour d'inventaire ou un avis de retrait de la liste des items en promotion. Pour l'affichage, on utilise une ligne par item.
Le fichier ventes.txt contient sur la première ligne le nombre d'items à traiter et sur chaque ligne suivante, le nom de l'item, le nombre d'items en inventaire et le nombre d'items vendus. Ces informations sont séparées par une tabulation ('\t') et chaque ligne se termine par le caractère fin de ligne (↵). Le nom de l'item ne contient que des caractères alphanumériques.

Question 4

```
int main()
 {
   ifstream fichierVente;
   string item;
   float nbItemInventaire, nbItemVendu;
   int nombreItem;

   fichierVente.open("ventes.txt");
   if (!fichierVente.fail())
   {
        fichierVente >> nombreItem;
        if (!fichierVente.eof())
        {
             fichierVente >> item >> nbItemInventaire >> nbItemVendu;
               while(!fichierVente.eof())
               {
                    if ((nbItemVendu /nbItemInventaire *100) < 17)
                          cout << "retrait de l'item " << item <<endl;
                    else if ((nbItemVendu /nbItemInventaire *100) > 37 )
                           cout << "demande d'ajustement pour l'item "
                                <<item <<endl;
                    else
                         ;
                    fichierVente >> item >> nbItemInventaire
                                 >> nbItemVendu;
               }
               fichierVente.close();

        }
        else
             cout<< "fichier vide"<<endl;
   }
   else
         cout << "le fichier n'existe pas"<<endl;
      return 0;
 }
```