



POLYTECHNIQUE
MONTRÉAL

Questionnaire examen final

INF1005C

Sigle du cours

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

Réservé

Sigle et titre du cours		Groupe	Trimestre
INF1005C – Programmation procédurale		Tous	20141
Professeur		Local	Téléphone
Martine Bellaïche, responsable François-Raymond Boyer		A-201	
Jour	Date	Durée	Heures
Samedi	26 avril 2014	2 h 30	9 h 30 – 12 h

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"/> Toutes <input type="checkbox"/> Non programmable	Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.

Directives particulières
L'indentation des programmes doit être faite correctement, mais vous n'avez pas à suivre le guide de codage. Vous pouvez utiliser le verso des pages pour votre brouillon. Il est inutile de mettre des commentaires.

Important	Cet examen contient <input type="text" value="4"/> questions sur un total de <input type="text" value="4"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="40"/> %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

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

Question 1 Représentation des nombres**(2 points)**

- a) (0,5 point) Effectuez l'opération suivante en complément à deux sur 8 bits. Indiquez s'il y a débordement, et pourquoi.

$$\begin{array}{r} 11010101 \\ + 10110111 \\ \hline \end{array}$$

Réponse : **110001100** Débordement? **non** Pourquoi? **l'addition de deux négatifs donnent un négatif.**

- b) (1,5 point) Rappel : le type `float` est représenté en IEEE 754 sur 32 bits, qui utilise 8 bits pour l'exposant, avec excès de 127. Donnez en binaire chaque partie de la représentation en `float` de la valeur décimale **01111111 + 5**.

Réponse : Signe : **0** Exposant : **0000 0001 = 1000 0000** Mantisse : **010 0000 0000 0000 0000 0000**

Question 2 Comportement des variables et paramètres**(3 points)**

Pour chaque sous-question, donnez l'affichage obtenu après l'exécution du programme.

- a) (1,5 point)

```
int f(int x, int& y)
{
    int t = x + y;
    x = x + 4;
    y = y + 8;
    return t;
}

int main()
{
    int a = 1;
    int b = 2;
    int c = f(a, b);
    cout << a << ' ' << b << ' ' << c;
}
```

Réponse : **1 10 3**

- b) (1,5 point)

```
struct S {
    int a, b;
};

int main()
{
    S a;
    char* b = new char[10];
    cout << sizeof(a) << ' ' << sizeof(b) << ' ' << sizeof(b[0]);
}
```

Réponse : **8 (car deux int de 4 octets chaque) 4 (4 octets pour tout pointeur) 1 (char 1 octets)**

Question 3 Vrai ou faux

(2 points)

Pour chaque sous-question, cochez vrai ou faux. Il est préférable de ne pas répondre à une sous-question que de cocher incorrectement.

V F

- a) Une variable de type `int` à laquelle on n'a fait aucune affectation contient la valeur zéro. ☐ ☒
- b) Soit un fichier binaire `istream fichBin`, on peut lire ce fichier jusqu'à la fin en utilisant la boucle
- ```
while (!ws(fichBin).eof())
```
- ☐ ☒
- c) L'entête de fonction suivante est valide : `int maFonction(istream fichier)` ☐ ☒
- d) Un type `struct` peut contenir un tableau de `string`. ☒ ☐

## Question 4 Programmation de fonctions

**(13 points)**

Nous voulons faire un programme permettant d'afficher les noms des personnes dont c'est la fête. Vous devez suivre le principe DRY (« don't repeat yourself »), c'est-à-dire que vous ne devez pas répéter de code entre les sous-questions, vous devez plutôt appeler les fonctions écrites dans les autres sous-questions si opportun. Vous devez aussi éviter les copies inutiles de structures et tableaux lors du passage de paramètres.

Soient les structures suivantes :

```
const int MAX_PERSONNES = 10;
struct Date {
 int annee, mois, jour;
};

struct Personne {
 string nom;
 Date naissance;
};

struct ListePersonnes {
 Personne personnes[MAX_PERSONNES];
 int nombreDePersonnes;
};
```

- a) (1,5 point)** Écrivez une fonction `estFete`, qui prend en paramètres la date de naissance d'une personne ainsi que la date actuelle, et détermine si c'est la fête de la personne, c'est-à-dire que les deux dates ont les mêmes mois et jours. La fonction doit retourner 'vrai' dans le cas où c'est la fête.

```
bool estFete(const Date& dateActuelle, const Date& dateNaissance) {

 return (dateActuelle.mois == dateNaissance.mois) &&
 (dateActuelle.jour == dateNaissance.jour);

}
```

- b) (1,5 point)** Écrivez une fonction `afficherFetes`, qui prend en paramètres une liste de personnes ainsi que la date actuelle, et qui affiche les noms de toutes les personnes dont c'est la fête.

```
void afficherFetes(const ListePersonnes& tous, const Date& dateActuelle) {
 for(int i = 0; i < tous.nombreDePersonnes; i++) {
 if (estFete(dateActuelle, tous.personnes[i].naissance)) {
 cout << tous.personnes[i].nom << endl;
 }
 }
 }
```

```
}
```

- c) (2 points)** Écrivez une fonction `ajouterPersonne`, qui prend en paramètres une liste de personnes ainsi qu'une personne, et ajoute la personne à la fin de la liste de personnes sans vérifier la capacité du tableau.

```
void ajouterPersonne(ListePersonnes tous, const Personne& unePersonne) {
 tous.personnes[tous.nombreDePersonnes] = unePersonne;
 tous.nombreDePersonnes++;
 }
```

```
}
```

- d) (1,5 point)** Soit une variable « `ListePersonnes listePersonnes;` » déjà existante, écrivez les instructions nécessaires pour ajouter « Alan Turing », né le 23 juin 1912, à cette liste de personnes.

```
ListePersonnes listePersonnes;
Personne unePersonne = { "Alan Turing", {1917, 6, 23}};
ajouter (listePersonnes, unePersonnes);
```

Text

Nous voulons maintenant permettre un nombre de personnes sans maximum fixé à l'avance.

- e) (1 point)** Indiquez comment modifier `ListePersonnes` pour que le tableau soit alloué dynamiquement, avec une certaine capacité, pour pouvoir le réallouer lorsqu'il est plein, mais pas à chaque ajout. Votre modification ne devrait pas affecter les fonctions que vous avez écrites ci-dessus.

```
struct ListePersonnes {
```

|                  |                           |
|------------------|---------------------------|
| <b>Personnes</b> | <b>Personnes;</b>         |
| <b>int</b>       | <b>nombreDePersonnes;</b> |
|                  | <b>int capacite;</b>      |

```
};
```

- f) (3 points)** Écrivez une fonction `augmenterCapacite`, qui prend en paramètres une liste de personnes ainsi qu'une nouvelle capacité, et qui réalloue le tableau de personnes pour avoir la nouvelle capacité en conservant toutes les personnes actuellement dans la liste. Vous n'avez pas à vérifier que la nouvelle capacité est suffisante pour le nombre de personnes.

```
augmenterCapacite(ListePersonne & tous, const int nouvCap) {
```

```
 Personne * nouveau = new Personnes[nouvCap];
```

```
 for (int i = 0; i < tous.nombreDePersonnes; i++) {
```

```
 nouveau[i] = tous.personnes[i]; }
```

```
 delete [] tous.personnes;
```

```
 tous.personnes = nouveau;
```

```
 tous.capacite = nouvCap;
```

```
}
```

Nous voulons maintenant utiliser un fichier binaire pour conserver une liste de personnes.

- g) (2,5 points)** Soit un fichier binaire qui contient des personnes (de type `Personne` modifié). Indiquez comment modifier la structure pour qu'elle puisse être utilisée dans un fichier binaire. Écrivez les instructions pour lire la 5<sup>e</sup> personne qui se trouve dans le fichier « `personnes.bin` », la placer dans la variable `personne`. Il faut ouvrir le fichier sans vérifier l'échec de l'ouverture.

Modification à `Personne` : **`struct Personne {char nom[30]; Date naissance;}`**

Instructions pour lire la 5<sup>e</sup> personne :

```
Personne personne;
```

```
 ifstream lecture("personne.bin", ios::binary);
```

```
 lecture.seekg(4*sizeof(Personne), ios::beg);
```

```
 lecture.read((char*)&personne, sizeof(personne));
```