Examen Final C++

Structures de contrôle

1. Ecrire une fonction qui, étant donné deux entiers x et y calcule x^y (x puissance y). Par exemple, pour x=3 et y=4 votre fonction doit retourner 81 (=3*3*3*3). **(2 points)**
2. Ecrire une fonction qui calcule la « semi-factorielle » d'un nombre entier. La semi-factorielle de *n* peut être calculée en multipliant tous les entiers impairs positifs inférieurs à *n*.
   Par exemple :
   semif(8) = 7*5*3*1 = 105 et
   semif(11) = 9*7*5*3*1 = 945. **(2 points)**

Tableaux

3. Ecrire une fonction qui décide si un tableau d'entiers contient un élément majoritaire. **(3 points)**
   Indications : Votre fonction aura comme arguments un tableau d'entiers ainsi que sa taille ***s***. S'il y a un nombre dans le tableau qui apparaît plus que ***s/2*** fois, votre fonction doit retourner la position (l'indice) où ce nombre apparaît pour la première fois. Si un tel nombre n'existe pas, votre fonction doit retourner -1.
   Exemple :
   Pour le tableau [5, 6, 7, 7, 7] votre fonction doit retourner 2,
   pour le tableau [5, 6, 7, 7, 8] elle doit retourner -1.

Pointeurs et gestion de la mémoire

4. Qu'affiche le programme suivant ? **(1 point)**

```
int l[ ] = {3,3,-1,1,-2};
int *p = l;
int *q;
int i;
for(i=0;i<5;i++){
q = p + (*p);
cout << *q << endl;
p++;
}
```

5. Considérez le programme suivant:

```
void flip(int *A, int s)

{
int *tmp = new int [s];
int i;
```

```
for(i=0;i<s;i++) tmp[i] = A[s-i-1];
for(i=0;i<s;i++) A[i] = tmp[i];
}
int main( )
{
int B = {1,2,3,4};
int C = {3,4,5,6};
int i;
flip(B,4);
flip(C,4);
for(i=0;i<4;i++) { cout << B[i] << “,” << C[i] << endl; };
}
```

Quel va-t-il afficher? Est-ce qu'il y a des erreurs? Si oui, comment les corriger? **(2 points)**

Listes chaînées

6. Ecrire une fonction qui, étant donné un pointeur sur la tête d'une liste chaînée, supprime un sur chaque deux éléments de cette liste. Plus précisément, il faudra supprimer le deuxième, le quatrième, le sixième, etc. élément, s'ils existent. **(3 points)**

Exemple : Pour la liste des éléments 1,2,3,4,5, votre programme doit supprimer 2,4, et devrait produire la liste 1,3,5. Assurez-vous que la mémoire pour les nœuds supprimés a été correctement libérée.

Note: Utilisez la définition suivante pour la structure liste chaînée.

```
struct Node { int data; struct Node *next; };
```

Classes

7. Ecrire une classe **Time.** Chaque objet de type **Time** stockera un certain nombre d'heures, de minutes et de secondes (tous des entiers).

On vous demande de proposer : **(3 points)**

1. La définition de la classe **Time**
2. Un constructeur sans arguments et un constructeur qui prend trois entiers.
3. Une méthode affiche( ), qui affiche les champs d'un objet Time.
4. Un opérateur d'addition qui permet d’additionner deux objets **Time**.

NB : Rappelons que les secondes et les minutes ne doivent jamais avoir une valeur >60.

Exemple : 75 secondes doit être convertie en 1 minute et 15 secondes.

NB : Cette information doit être prise en compte dans le constructeur et dans l’opérateur d'addition.

8. Considérons la classe suivante

```
class A{
int *data;
public:
A( ) { cout <<"Const"<<endl ; data = new int; };
~A( ) { cout <<"Dest"<<endl; delete data; };
};
```

• Qu'affiche le programme suivant? **(1 point)**

```
int main() { A a1,a2,a3; }
```

• Qu'affiche le programme suivant? **(1 point)**

```
void f(A a) { cout <<"f" << endl;}
int main( ) { A a1; f(a1); f(a1); f(a1); }
```

Héritage

9. Qu'affiche le programme suivant? **(2 points)**

```
class A{
protected:
int data;
public:
A(int d=0) { data=d; };
void f( ) { cout << "A::f:" << data << endl; };
virtual void g( ) { cout << "A::g:" << data << endl; };
};
class B: public A{

int moredata;
public:
B(int m=0) { moredata = m;};
void f( ) { cout << "B::f:" << data << "," << moredata << endl; };
void g( ) { cout << "B::g:" << data << "," << moredata << endl; };
};
int main()
{
A a(1); B b(2); A *p; p = &a; p->f(); p->g(); p = &b; p->f(); p->g();
}
```

Templates

10. Donner une version générique du programme de l'exercice 3 (élément majoritaire). **(1 point)**