

Nom : _____

Prénom : _____

Matricule : _____

L'examen est sur 100 points et comprend 4 questions. Un résumé du C++ est inclus sur cette page.

Les opérateurs C++ (en ordre décroissant de priorité)

<i>Catégorie d'opérateurs</i>	<i>Opérateurs</i>
fonction, tableau, membre de structure, pointeur sur un membre de structure	() [] . ->
opérateurs unaires	- ++ -- ! ~ * & sizeof (type)
multiplication, division, modulo	* / %
addition, soustraction	+ -
opérateurs binaires de décalage	<< >>
opérateurs relationnels	< <= > >=
opérateurs de comparaison	== !=
et binaire	&
ou exclusif binaire	^
ou binaire	
et logique	&&
ou logique	
opérateur conditionnel	?:
opérateurs d'affectation	= += -= *= /= %= &= ^= = <<= >>=
opérateur virgule	,

Déclaration d'une classe

```
class MaClasseDerivee: public MaClasseDeBase {  
public:  
    MaClasseDerivee();  
    MaClasseDerivee(const MaClasseDerivee&);  
    ~MaClasseDerivee();  
    const MaClasseDerivee& operator=(const MaClasseDerivee&);  
};
```

Flots

```
ofstream, ifstream, cin, cerr, cout  
cin >> P; //lecture  
cout << P; // écriture
```

Question 1. Technique (25 points)

A. Expliquez aussi clairement que possible mais brièvement ce qu'est X dans les expressions suivantes. Vous ne devriez pas utiliser plus d'une phrase par expression. (20 points)

double X; (2 points)

double * X; (2 points)

double * X[10]; (2 points)

double (*X)[10]; (2 points)

double X(10); (2 points)

double X[10]; (2 points)

double& X; (2 points)

double*& X; (2 points)

double p1 = 1.0;
double *p2 = &p1;
double*& X = p2; (2 points)

MaClasse Y;
MaClasse& X = Y; (2 points)

B. Qu'est-ce qui s'affichera à l'écran après l'exécution de ce code? (5 points)

```
for (int b = 2; b <= 2; b++) cout << b++; cout << b;
```

Question 2. Utilisation d'un tableau (25 points)

Il n'est pas rare en C++ qu'on veuille sécuriser l'accès à un pointeur. En voici un exemple:

```
template <class TYPE>
class MonTableau {
public:
    MonTableau(const int N) {
        tableau = new TYPE[N];
    }

    ~MonTableau() {
    }

    const TYPE operator[](const int indice) {
        return tableau[indice];
    }

protected:
    TYPE * tableau;
    MonTableau(const MonTableau&);
};

void main() {
    MonTableau mt(10);
    mt[2] = 3;
}
```

A. Trouvez deux erreurs dans ce code qui empêcheront la compilation et corrigez (dans le code). (10 points)

B. Un programmeur vient vous consulter parce qu'après avoir corrigé les deux erreurs précédentes, le code suivant pose toujours un problème. (Expliquez en une phrase.) (5 points)

```
const int N = 1024*1024*10;
for(int k = 0; k < 1000; k++) {
    MonTableau mt(N);
}
```

C. Est-ce que cette classe est sécuritaire? Proposez au moins une modification qui pourrait rendre cette classe plus sécuritaire. (On peut supposer que l'opérateur « = » et le copieur ne sont pas utilisés.) (10 points)

Question 3. Utilisation d'une Pile (25 points)

D'une part, on désire dériver la classe Pile ci-après de telle manière que l'on puisse connaître en tout temps le nombre d'éléments présents dans la pile. D'autre part, on désire aussi faire en sorte que la taille de la Pile ne dépasse jamais 1 000 000 éléments et qu'en cas de débordement, l'ajout d'un élément remplace simplement l'élément au-dessus de la pile.

```
template <class TYPE>
class Pile {
public:
    Pile();
    virtual ~Pile();
    virtual void empiler(const TYPE&);
    virtual TYPE depiler();
    virtual void vider();
    bool estVide() const;
```

```
protected:    /*...*/
};
```

Écrivez votre classe ici.

```
template <class TYPE>
class MaPile: public Pile<TYPE> {
public:
    MaPile() { //mettre votre code ici

    }
    void empiler(const TYPE& t) { //mettre votre code ici

    }
    TYPE depiler() { //mettre votre code ici

    }
    void vider() { //mettre votre code ici

    }
    int nombreDElements() { //mettre votre code ici

    }
private: //mettre votre code ici

};
```

Question 4. Utilisation d'une file (25 points)

La norme vidéo H.261 utilise une file afin de respecter les limites de la bande passante disponible (un multiple de 64 kbps). Vous devez coder une classe dont la méthode « tick » sera appelée 1000 fois par secondes (aux millisecondes), la file contient des objets « Image » munis d'une méthode « int taille() » qui donne la taille en octets de chaque image. La méthode tick doit donner soit un pointeur « NULL » soit un pointeur sur une image, mais il faut qu'en moyenne, vous ne dépassiez pas une utilisation de la bande passante de 240 octets par seconde.

Attention! Vous devez tenir compte du cas où la file serait vide!!!

La classe file est une classe générique munie des opérations suivantes :

```
void enfiler(const TYPE&);  
TYPE defiler();  
bool estVide() const;  
void vider();
```

```
class Video {  
    Video(file<Image> * f) { // mettre votre code ici  
  
  
    }  
    Image * tick() { // mettre votre code ici  
  
  
    }  
private: // mettre votre code ici  
  
  
};
```