

Structures de données et algorithmes fondamentaux Examen final Durée : 120 minutes Tous les documents sont autorisés

Nom et prénom : _____

Question :	1	2	3	4	5	6	7	Somme
Barème :	4	2	2	5	2	3	2	20
Points :								

1. (4 points) Donner les réponses correctes.

- (a) La taille d'un vecteur ne peut pas être changée après création.
 vrai faux
- (b) La déclaration "vector<int, int> vec" crée un vecteur de vecteurs d'entiers.
 vrai faux
- (c) L'accès `vec.at(i)` à l'élément à l'indice `i` du vecteur `vec` lance une exception si `i` est strictement négatif.
 vrai faux
- (d) Bien que ça ne soit pas conseillé, une donnée membre peut être publique.
 vrai faux
- (e) Toute classe doit avoir au moins une fonction membre.
 vrai faux
- (f) Sous l'hypothèse que les éléments d'un vecteur d'entiers sont ordonnés de manière *décroissante*, la recherche d'un entier donné peut être faite en temps $O(\log n)$ où n est la taille du vecteur.
 vrai faux
- (g) L'algorithme du tri par sélection prend temps $O(n^2)$ où n est la taille du vecteur à trier.
 vrai faux
- (h) Il n'est pas possible pour une fonction de s'appeler elle-même.
 vrai faux

2. (2 points) Écrire une fonction qui prend un vecteur de strings et qui renvoie un élément de longueur maximale. S'il y a plusieurs éléments avec la longueur maximale, vous pouvez choisir librement un de ces éléments. Vous pouvez supposer que le vecteur n'est pas vide.

3. (2 points) Écrire une fonction qui prend un vecteur d'entiers et qui supprime le dernier élément (s'il existe).

4. (5 points) Écrire une classe Phrase décrite comme suit. Elle doit contenir une suite de strings et des fonctions pour effectuer les opérations suivantes :
- Rajouter un string à la fin de la suite de strings.
 - Vérifier si la phrase est une question, c'est-à-dire si le dernier string dans la suite finit avec le caractère '?'.
 - Calculer la longueur totale de la phrase, c'est-à-dire la somme des longueurs des strings dans la suite.

Spécifier le fichier (.h ou .cpp) pour les différentes parties de code que vous écrivez. Il n'est pas nécessaire de donner les instructions using, #include, #ifndef, #define et #endif.

5. (2 points) Écrire un programme qui laisse l'utilisateur saisir une suite d'entiers jusqu'à ce qu'il mette l'entier 42. Ensuite, le programme affiche tous les entiers saisis (42 inclus) dans l'ordre inverse, en omettant les entiers nuls.

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
}
```

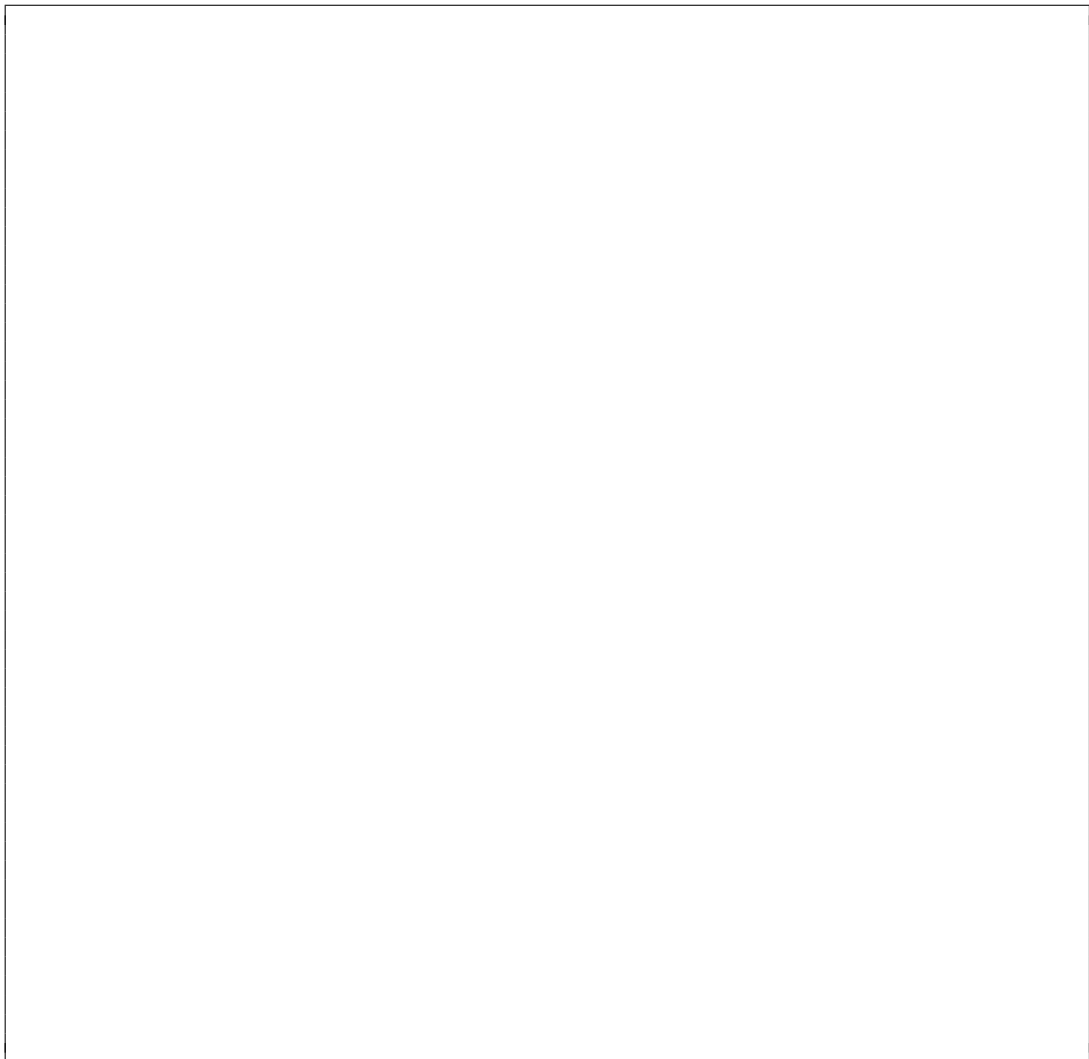
6. (3 points) Soit donnée la classe suivante :

```
class vec_trie {  
private:  
    vector<int> elems;  
  
public:  
    int rang(int x) const;  
    void rajouter(int x);  
};
```

Pour accélérer la recherche d'un élément, la fonction membre rajouter garantit que le vecteur elems reste toujours trié.

- (a) Compléter l'implémentation de la fonction membre rang pour trouver le nombre d'éléments du vecteur qui sont inférieurs ou égaux à l'entier x. Faire attention à la complexité en temps de votre algorithme.

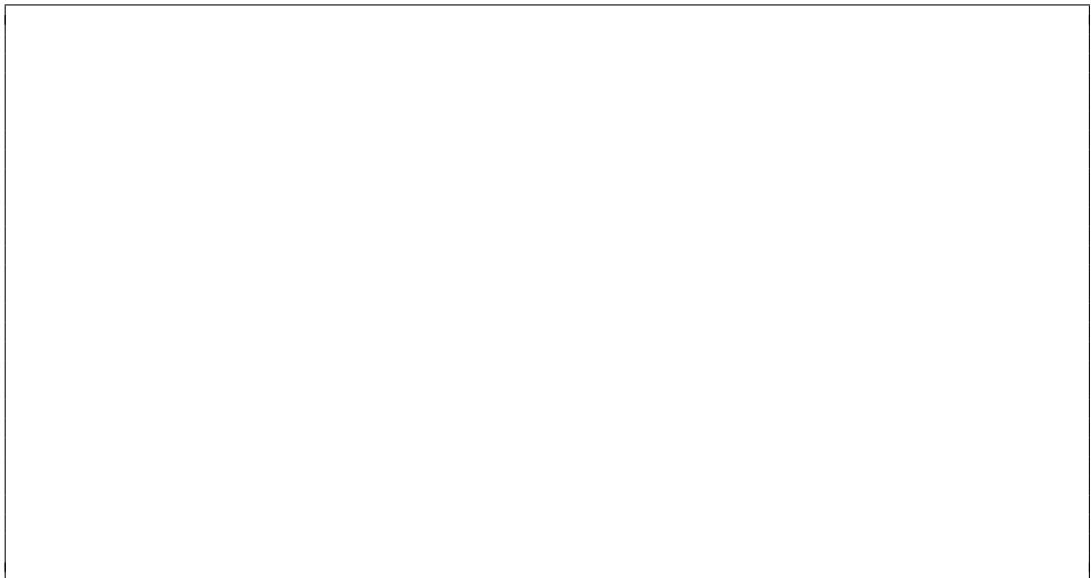
```
int vec_trie::rang(int x) const  
{
```





}

(b) Quelle est la complexité en temps de votre algorithme? Pourquoi?



7. (2 points) Considérer les fonctions suivantes :

```
void f(int n)
{
    for(int i = 0; i < n; ++i)
    {
        cout << ".";
        ++i;
    }
}

void g(int n)
{
    for(int i = 0; i < n; ++i)
    {
        for(int j = 0; j < n-i; ++j)
        {
            cout << ".";
        }
    }
}
```

(a) Quel est le nombre exact d'itérations de la boucle dans la fonction f ? Expliquer.

- (b) Quel est le nombre d'itérations de la boucle intérieure dans la fonction g exprimé en notation de Landau ? Expliquer.

