

Cours d'introduction à l'informatique

Examen d'août 2024

Livres fermés. Durée : 3 h 30

Veillez répondre aux questions sur des feuilles séparées sur lesquelles figurent votre nom, prénom et matricule. Les réponses doivent être justifiées. Tous les programmes et fragments de code demandés doivent être rédigés en langage C.

1. (a) Écrire une fonction prenant en arguments un tableau t d'entiers signés supposés triés par ordre croissant, ainsi que la taille n de ce tableau, et retournant le nombre de valeurs différentes parmi les éléments de t . Par exemple, si t contient $[-1, -1, 4, 5, 5]$, alors la fonction doit retourner 3. Si $n = 0$, la fonction doit retourner 0.
(b) Par la méthode des invariants, démontrer que la valeur retournée par cette fonction est correcte.
(c) Calculer la complexité en temps de cette fonction.
2. Le *triangle d'Euler* est une construction mathématique constituée d'une séquence de lignes de longueur croissante : pour $i = 1, 2, \dots$, la ligne d'indice i contient i nombres entiers $A_{i,0}, A_{i,1}, \dots, A_{i,i-1}$ définis de la façon suivante :

— Le premier et le dernier élément de chaque ligne sont égaux à 1 :

$$\forall i > 0 : A_{i,0} = A_{i,i-1} = 1.$$

— Les autres éléments sont déterminés par l'équation

$$\forall 0 < j < i - 1 : A_{i,j} = (i - j)A_{i-1,j-1} + (j + 1)A_{i-1,j}.$$

- (a) Écrire un programme qui affiche à l'écran les N premières lignes du triangle d'Euler, où $N \geq 1$ est une constante définie au début de ce programme.

Par exemple, pour $N = 6$, le programme doit produire l'affichage suivant :

```
1
1 1
1 4 1
1 11 11 1
1 26 66 26 1
1 57 302 302 57 1
```

- (b) Calculer la complexité en temps et en espace du programme obtenu au point (a).

3. (a) Décrire le plus simplement possible ce que calcule la fonction C suivante :

```
unsigned f(unsigned i, unsigned j)
{
    if (i < j)
        return f(j, i);
    if (j == 0)
        return 0;
    return i + f(i, j-1);
}
```

- (b) Quelle est la complexité en espace de cette fonction ?
- (c) Écrire une fonction C réalisant exactement la même opération que la fonction du point (a), mais sans effectuer d'appel récursif.
4. Une application de gestion de notes d'étudiants représente ces étudiants comme les éléments d'une liste simplement liée. Chacun de ces éléments est composé du matricule d'un étudiant, formé par 8 caractères, et d'un pointeur vers l'élément suivant dans la liste, ce pointeur étant vide pour le dernier élément. La liste liée retient uniquement un pointeur vers son premier élément (qui est vide si cet élément n'existe pas), ainsi que le nombre d'éléments qu'elle contient.
- (a) Écrire un fragment de code C définissant deux types structurés : le premier pour un élément de liste liée représentant un étudiant, et le second pour une liste simplement liée représentant une séquence d'étudiants.
- (b) Écrire une fonction prenant en arguments un pointeur vers la représentation d'une liste liée ℓ d'étudiants, et un tableau de caractères contenant le matricule m d'un étudiant supplémentaire. Si le matricule m n'est pas déjà présent dans ℓ , alors la fonction doit ajouter l'étudiant correspondant à la fin de cette liste et retourner 0. Sinon, la fonction doit laisser ℓ inchangée et retourner -1 .
- Notes* : Les types des éléments de la liste liée et de la liste liée elle-même doivent correspondre à votre réponse au point (a). Vous pouvez programmer des fonctions ou des types de données supplémentaires si votre solution le nécessite.
- (c) Écrire une fonction prenant en argument un pointeur vers la représentation d'une liste liée, compatible avec votre réponse au point (a), et libérant l'ensemble de la mémoire allouée à cette liste et à ses éléments.