

Durée : 1h. Aucun document n'est autorisé. Les téléphones doivent être rangés.

Lorsqu'une question vous demande la complexité d'un algorithme, vous devez donner son ordre de grandeur (notation Θ).

Ce contrôle n'est pas anonyme. Inscrivez sur la copie vos nom, prénom et numéro étudiant.

Question 1 Soit l'algorithme g :

```

Algorithme :  $g(d \text{ n : entier}) : \text{entier}$ 
Données :  $n \in \mathbb{N}^*$ 
Résultat : Renvoie l'entier ???
Variables :  $R, V, i \in \mathbb{N}$ ;
début
   $r \leftarrow 1; p \leftarrow 1; i \leftarrow 1;$ 
  tant que  $p < n$  faire
     $i \leftarrow 2 + i; r \leftarrow r + i; p \leftarrow p + 1;$ 
  fin tq
  renvoyer  $r$ ;
fin

```

- (0,5 pt)** Montrez que l'algorithme g s'arrête en donnant une expression entière qui décroît strictement à chaque itération.
- (0,5 pt)** Faites la trace de l'exécution de $g(5)$ en indiquant sous forme d'un tableau les valeurs des variables p, i, r à chaque itération.
- (2,5 pts)** Quelle propriété invariante relie les valeurs des variables i et p ? Montrez cet invariant.
- (2,5 pts)** Quelle propriété invariante relie les valeurs des variables r et p ? Montrez cet invariant.
- (1 pt)** Que calcule l'algorithme g ? Prouvez-le.

Question 2 (a) **(4 pts)** Écrivez un algorithme qui étant donnés 2 entiers positifs a et b , $b \neq 0$, calcule le quotient de la division entière de a par b qui est non nul.

Ex : le quotient de la division de 17 par 3 est 5 ; le quotient de la division de 17 par 7 est 2.

Les seules opérations arithmétiques que vous pouvez utiliser sont l'addition et la soustraction. Vous pouvez utiliser les relations de comparaison.

```

Algorithme : quotient( $d \text{ a : entier, d b : entier, r q : entier}$ )
Données :  $a \in \mathbb{N}, b \in \mathbb{N}^*$ 
Résultat :  $q$  est le quotient de la division entière de  $a$  par  $b$ , c'est à dire  $q$  vérifie  $q \times b \leq a < q \times (b + 1)$ 

```

- (2,5 pts)** Faites la preuve complète de votre algorithme :
 - donnez sans le prouver un argument attestant que votre algorithme s'arrête
 - donnez sans le prouver un invariant significatif de votre itération.
 - montrez que cet invariant permet d'établir que le résultat calculé par votre algorithme est correct.

Question 3 Un tableau de nombres est dit *bitone* s'il est constitué d'une séquence strictement croissante suivie d'une séquence strictement décroissante. Chaque séquence peut éventuellement être vide. Plus précisément T est un tableau *bitone* si et seulement si il existe un indice $m \in [0, \text{taille}(T)[$ vérifiant :

$$(\forall i, j \text{ tels que } 0 \leq i < j \leq m, T[i] < T[j]) \text{ et } (\forall k, l \text{ tels que } m \leq k < l < \text{taille}(T), T[k] > T[l])$$

Ex :

5	6	8	9	8	4
---	---	---	---	---	---

,

2	5	6	8
---	---	---	---

 et

9	7	6	4	3
---	---	---	---	---

 sont 3 tableaux bitones.

1	9	7	6	8
---	---	---	---	---

 et

5	3	4
---	---	---

 ne sont pas des tableaux bitones.

- (5 pts)** Écrivez un algorithme qui calcule **par dichotomie** la valeur maximum des éléments d'un tableau bitone.

Par exemple si la donnée est le tableau

5	6	8	9	8	4
---	---	---	---	---	---

 le résultat est 9.

- (1,5 pts)** Vous donnerez, sans le prouver, un invariant significatif (permettant d'attester que votre algorithme est correct) vérifié par votre itération.

Quelle est la complexité de votre algorithme?