

Nom : _____

Prénom : _____

- Durée : 1 heure 30. Documents interdits, calculatrices interdites.
- Écrivez toutes les réponses directement sur le sujet. Si vous n'avez pas suffisamment de place, écrivez au dos d'une feuille en le **précisant dans la question**.

1 Divisions (4 points)

1. 2 points Ecrire en pseudo-code avec les conventions de votre choix une fonction *divise* prenant deux entiers a et b en paramètres et retournant *vrai* si et seulement si $a|b$. Vous supposerez que vous avez à votre disposition une fonction *modulo*(x, y) qui retourne le reste de la division entière de x par y . On supposera que a est différent de 0.

Solution:

```
Fonction divise( $a, b$ )  
| retourner modulo( $b, a$ )  $\neq 0$   
FIN
```

2. 2 points Ecrire en pseudo-code avec les conventions de votre choix une fonction *estPremier* prenant un entier a en paramètre et retournant *vrai* si et seulement si a est un nombre premier. Vous utiliserez la fonction *divise* et réfléchirez particulièrement à la condition de sortie de la boucle. On supposera que a est strictement positif.

Solution:

```
Fonction estPremier( $a$ )  
|  $i \leftarrow 2$   
| tant que  $i^2 \leq a$   
| | si divise( $i, a$ ) alors  
| | | retourner vrai  
| | fin  
| |  $i \leftarrow i + 1$   
| fin tant que  
| retourner faux  
FIN
```

2 PGCD (5 points)

1. 3 points Ecrire en pseudo-code avec les conventions de votre choix une fonction *pgcd* retournant le *pgcd* de deux entiers relatifs *a* et *b* passés en paramètre. Vous écrirez cette fonction de façon itérative (c'est à dire avec une boucle). Tout algorithme récursif sera considéré comme faux.

Solution:

```
Fonction pgcd(a, b)  
  | tant que b ≠ 0  
  |   | r ← modulo(a, b)  
  |   | a ← b  
  |   | b ← r  
  | fin tant que  
  | retourner a  
FIN
```

2. 2 points Donner la version récursive de la fonction précédente.

Solution:

```
Fonction pgcd(a, b)  
  | si b = 0 alors  
  | | retourner a  
  | fin  
  | retourner pgcd(b, a mod b)  
FIN
```

3 Changement de base (11 points)

1. 5 points Ecrire en pseudo-code avec les conventions de votre choix une fonction non récursive $enBase2(x, t)$ prenant en paramètres un entier x et un tableau t (supposé de taille suffisante). Vous devez convertir x en base 2 en plaçant dans t les chiffres de x . Ces chiffres seront placés à l'envers (celui de poids fort à la fin). Cette fonction retourne le nombre d'éléments significatifs du tableau. Vous supposerez que vous avez à votre disposition une fonction $modulo(x, y)$ qui retourne le reste de la division entière de x par y .

Solution:

```
Fonction  $enBase2(x, t)$   
  | si  $x = 0$  alors  
  |   |  $t[1] \leftarrow 0$   
  |   | retourner 1  
  |   fin  
  |  $k \leftarrow 0$   
  | tant que  $x \neq 0$   
  |   |  $k \leftarrow k + 1$   
  |   |  $t[k] \leftarrow modulo(x, 2)$   
  |   |  $x \leftarrow x/2$   
  |   fin tant que  
  | retourner  $k$   
FIN
```

2. 6 points Ecrire en pseudo-code avec les conventions de votre choix une fonction $enBase10(t)$ prenant en paramètre un tableau t à n éléments contenant la représentation en base 2 d'un nombre. Cette fonction convertit le nombre contenu dans t en base 10 et le retourne. On supposera que le chiffre de poids fort se trouve à la fin du tableau t . Vous utiliserez pour reconstituer le nombre en base 10 la règle de Horner : si par exemple vous devez convertir 11001 (donc le tableau 10011) en base 10, vous utiliserez la formule $((((1)2 + 1)2 + 0)2 + 0)2 + 1$.

Solution:

```
Fonction enBase10(t)
| res ← 0
| k ← n
| tant que k > 0
| | res ← 2 * res + t[k]
| | k ← k - 1
| fin tant que
| retourner res
FIN
```