Université Ibn Zohr
Faculté des Sciences d'Agadir
Département d'Informatique
A.U : 2018 / 2019

Examen de Session de Rattrapage d'Algorithmique 2
Filière : SMI3
Durée : 1h30 - Documents non autorisés

Important :
- **Lire tout l'énoncé** de l'examen avant de commencer à répondre.
- L'ordre de difficulté des exercices n'est pas nécessairement croissant.
- Le prêt de **tout matériel** entre étudiants est **formellement interdit**.
- Vous pouvez utiliser les fonctions prédéfinies vues au cours.

Exercice 1 : Preuve d'algorithme [5 points] [~ 25 minutes]
On considère l'algorithme suivant :

```
Entrées : un entier naturel x et un entier naturel y
Résultat : un nombre r
Variables r,s,t,w : Entier
Début
r ← 0
s ← x
Tant que s > 0 faire
  r ← r + y
  s ← s – 1
FinTantQue
w ← x – 1
t ← r
Tant que w > 0 faire
  r ← r + t
  w ← w – 1
FinTantQue
Fin
```

1. Calculer la valeur finale de r lorsque (x ; y) = (3; 2) et (x ; y) = (4; 3).
2. Justifier que l'algorithme se termine en précisant le variant de la première et la deuxième boucle TantQue.
3. Démontrer par récurrence que :
   - "$r_i = (x - s_i)y$" est un invariant de la première boucle TantQue.
   - "$r_i = (x - w_i)xy$" est un invariant de la deuxième boucle TantQue.
4. Que fait cet algorithme (valeur de la variable r à la fin) ?

Exercice 2 : [4 points][~ 20 minutes]
Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs, qui sont 1 et lui-même.
Exemples : 19 ; 31 ; 61
1. Ecrire une fonction qui prend en paramètre un entier naturel P et retourne 1 si P est un nombre premier et 0 sinon.
2. Ecrire une fonction **SPI** qui prend en paramètre un entier naturel N et retourne la Somme des nombres Premiers Inférieurs ou égal à N.
   Exemples : si N= 6 alors SPI(6) = 2 + 3 + 5 = 10
   si N = 7 alors SPI(7) = 2 + 3 + 5 + 7 = 17
3. Un nombre K est P-premier, si K est premier **ET** si SPI(K) est premier.
   Exemple : 7 est P-premier.
   Ecrire une fonction Ppremier qui prend en paramètre un entier naturel K et retourne 1 si K est P-premier et 0 sinon.
4. En justfiant votre réponse, calculer si les entiers suivants sont P-premiers ou pas :
   - N = 13 ; N = 16 ; N = 19 ; N = 27 ;

Exercice 3: [7 points][~ 30 minutes]

Voici un algorithme qui réalise la multiplication de 2 nombres entiers N et M en n'utilisant que des multiplications/divisions par 2 selon la conception suivante :

À chaque étape, N est divisé par 2 (division entière) et M est multiplié par 2. Si N est impair, la valeur de M est ajoutée au futur résultat. Si N est strictement positif, on s'arrête à N = 1.

Exemple: 321 * 457

```
N          M
321       457      N est impair donc futur résultat=457
160       914      N est pair donc on n'ajoute pas 914
 80      1828      N est pair donc on n'ajoute pas 1828
 40      3656      N est pair donc on n'ajoute pas 3656
 20      7312      N est pair donc on n'ajoute pas 7312
 10     14624      N est pair donc on n'ajoute pas 14624
  5     29248      N est impair donc futur résultat = 457 + 29248 = 29705
  2     58496      N est pair donc on n'ajoute pas 58496
  1    116992      N est impair donc résultat = 29705 + 116992 = 146697
```

1. **En justifiant votre réponse**, calculer le produit N*M en utilisant l'algorithme de multiplication par shifting ci-dessus pour les cas suivants : (une réponse non justifiée est fausse)
   - N = 21 , M = 5
   - N = 107, M = 3
   - N = 5 , M = 21
2. Ecrire une fonction itérative **prod_shift_iter** qui prend en paramètres deux entiers, respectivement, N et M et retourne leur produit. La fonction doit prendre en considération qu'un des deux paramètres peut être nul.
   N.B. Attention aux initialisations dans la fonction.
3. Quelle est la complexité de cette solution (nombre de divisions) ? Justifier votre réponse.
4. Ecrire une fonction **prod_shift_rec** la version récursive de la fonction **prod_shift_iter.**
5. Ecrire un programme qui lit deux entiers positifs N et M puis, pour calculer leur produit, fait appel à la fonction itérative **prod_shift_iter** de sorte à ce que le nombre d'itérations soit **minimal**.

Exercice 4: [4 points][~ 15 minutes]

On considère que :

- Chaque ligne d'un fichier texte est une chaine de caractères qui se termine par '\0'
- Une ligne ne peut pas être vide.

Ecrire un algorithme qui affiche les caractéristiques suivantes d'un fichier texte nommé « Exemple.txt » :

- Nombre de lignes,
- Nombre de mots,
- Nombre de caractères (y compris les espaces).