Dernière mise à jour	Informatique	Denis DEFAUCHY – <u>Site web</u>
18/03/2023 9 – Algorithmique		INT1 – Sujet

Note

Algorithmique

Nom:

Exercice 1: Terminaison - Correction

Prénom:

Soit l'algorithme suivant :

```
# Factoriel n, n>0

n = 10
k = n
s = n
while k>1:
    k -= 1
    s = s*k
print(s)
```

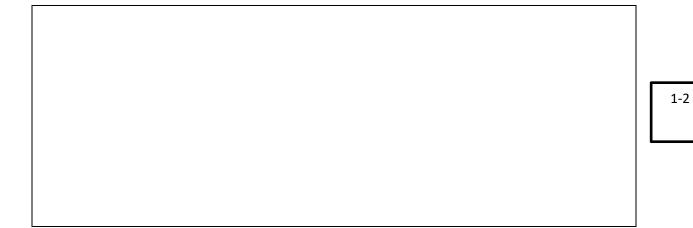
Il est censé calculer le factoriel :

$$n! = \prod_{i=1}^{n} i$$

Question 1: Comment prouve-t-on la terminaison d'un algorithme ?

1-1

Question 2: Vérifier la terminaison de cet algorithme



@ 080 BY NC SA

Dernière mise à jour	Informatique	Denis DEFAUCHY – <u>Site web</u>
18/03/2023	9 – Algorithmique	INT1 – Sujet

Question 3: Réaliser la correction (preuve) de cet algorithme

Algorithme	
Propriété	
Initialisation	
Transmission	
Sortie	

1-3

Dernière mise à jour	Informatique	Denis DEFAUCHY – <u>Site web</u>
18/03/2023	9 – Algorithmique	INT1 – Sujet

Exercice 2: Complexité

Question 1: Compléter le tableau suivant en indiquant pour chaque complexité la complexité équivalente la plus simple $\mathcal{C}(n)$ de la fonction f(n)

f(n)	C(n)
2 <i>n</i>	
$5n + n^2$	
2 ⁿ⁺³	
$4^{n} + 2^{n}$	
$\log_2 n$	

2-1

Dernière mise à jour	Informatique	Denis DEFAUCHY – <u>Site web</u>
18/03/2023	9 – Algorithmique	INT1 – Sujet

Question 2: Donner la complexité en temps des algorithmes suivants en justifiant brièvement le résultat. S'il y a lieu, préciser meilleur et pire des cas

2-2

N°	Programme	Complexité
1	<pre>n = 100 S = 0 for i in range(n): S += 1</pre>	O(n) On passe n fois dans la boucle
2	<pre>n = 100 S = 0 for i in range(n): for j in range(n): S += 1</pre>	$O(n^2)$ On exécute n fois une boucle de n calculs $n st n = n^2$
3	<pre># L étant une liste de n termes x = 10 if x in L: print('Oui') else: print('Non')</pre>	X in L est en O(1) dans le meilleur des cas et O(n) dans le pire des cas
4	n = 1000 while n>1: n = n/2	L'algorithme s'arrête dès que : $\frac{n}{2^t} < 1$ $\Leftrightarrow 2^t > n$ $\Leftrightarrow e^{t \ln 2} > n$ $\Leftrightarrow t \ln 2 > \ln n$ $\Leftrightarrow t > \frac{\ln n}{\ln 2}$ Donc : $O(\ln n)$