

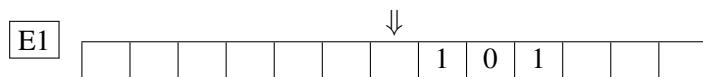
Spécialité NSI en terminale
Devoir surveillé 1

Exercice 1

On considère la table de transition d'une machine de Turing, donnée par le tableau suivant :

État	Lecture	Écriture	Déplacement	État suivant
E1	blanc	blanc	gauche	E2
E2	0	0	gauche	E2
	1	1	gauche	E2
	blanc	blanc	droite	E3
E3	0	blanc	gauche	Fin
	1	blanc	gauche	Fin

Le déplacement décrit par la table est celui du ruban. L'état de départ avec le nombre 101 écrit sur le ruban est le suivant :



1. Décrire les différentes étapes jusqu'à l'état final.
2. Quel est le nombre obtenu dans les cas suivants :
 - (a) s'il est écrit 1010 sur le ruban ;
 - (b) s'il est écrit 1001 sur le ruban.
3. Si le nombre écrit sur le ruban est l'écriture binaire de 17, quel est le résultat obtenu en décimal ?
4. Préciser quelle est l'opération mathématique réalisée.

Exercice 2

On considère la fonction f définie ci-dessous qui prend en paramètre un entier naturel n .

```
def f(n):
    nstr = str(n)
    s = 0
    for c in nstr:
        s = s + int(c)
    return s
```

1. Quelle est la valeur de $f(52431)$?
2. Ecrire une fonction g , version récursive de la fonction f .

Exercice 3

On considère une fonction récursive add qui additionne deux entiers naturels en ajoutant simplement une unité à chaque appel récursif.

```
def add(a, b):
    if b == 0:
        return a
    else:
        return add(a, b-1) + 1
```

1. Déterminer en fonction de b le nombre d'appels de la fonction `add` pour effectuer $a + b$.
2. Expliquer pourquoi la terminaison n'est assurée que dans le cas où b est un entier naturel.
3. Démontrer la correction de la fonction, autrement dit, prouver que l'expression `add(a, b)` a pour valeur la somme $a + b$.
4. Ecrire en suivant le même modèle une fonction récursive `mul` qui effectue la multiplication de deux nombres entiers naturels. Utiliser la fonction `add`; l'opérateur `+` n'est pas autorisé.
5. Ecrire une fonction récursive calculant a^b , avec des valeurs de b entières positives ou nulles. Utiliser la fonction `mul`; ici l'opérateur `*` n'est pas autorisé.

Exercice 4

On considère les deux fonctions `compte1` et `compte2` définies ci-dessous.

```
def compte1(n):
    if n >= 0:
        print(n)
        compte2(n-1)

def compte2(n):
    if n >= 0:
        compte1(n-1)
        print(n)
```

Le paramètre n est un entier naturel. Déterminer le résultat de l'appel `compte1(3)` puis celui de l'appel `compte2(3)`.