

Mathématiques Discrètes

feuille numéro 06

15 octobre 2018

Exercice 1

On donne le code PYTHON suivant :

```
def number_of_files_utf8_length(n):
    """ utf-8 maladroit
    compute recursively the term of
    index n of the number of bytes n string
    which are valid utf-8 string

    :param n: the index of the term
    :type n: int
    :rtype: int
    :UC: n is a non negative int. if not raise AssertionError
    :Examples:

    >>> number_of_files_utf8_length(1)
    128
    >>> number_of_files_utf8_length(0)
    1
    >>> number_of_files_utf8_length(5)
    57176752128
    """
    assert type(n)==int
    assert n>=0

    vect=[128,2048,65536,2097152]
    u=[1,128,18432,2686976]
    if n<=3:
        return u[n]
    else:
        return sum( [vect[i]*number_of_files_utf8_length(n-1-i)
                    for i in range(4)])
```

Q 1.1 Lire le code !

La fonction `number_of_files_utf8_length` implante le calcul d'une suite définie par récurrence.

Q 1.2 Précisez laquelle ?

On définit la suite a de la manière suivante : « a_n désigne le nombre d'appels à la fonction `number_of_files_utf8_length` effectués lors du calcul de $x(n)$ »

Q 1.3 Quelle est l'équation de récurrence de a_n ?

Exercice 2

On donne le code PYTHON commenté suivant :

```
def fibo(n):
    """ suite de Fibonacci maladroite
    compute recursively the term of
    index n of the Fibonacci sequence
```

```

:param n: the index of the term
:type n: int
:rtype: int
:UC: n is a non negative int. if not raise AssertionError
:Example:

```

```

>>> fibo(0)
0
>>> fibo(1)
1
>>> fibo(5)
5
>>> fibo(6)
8
"""
assert type(n)==int and n>=0
if n<=1:
    return n
else: #ici n>1 donc n>=2
    return fibo(n-1)+fibo(n-2)

```

Q 2.1 Lire le code!

La fonction `fibo` implante le calcul d'une suite définie par récurrence.

Q 2.2 Précisez laquelle? (pas son nom mais son équation de récurrence!)

On définit la suite a de la manière suivante : « a_n désigne le nombre d'appels à la fonction `fibo` effectués lors du calcul de `fibo(n)` »

Q 2.3 Quelle est l'équation de récurrence de a_n ?

On définit la suite b de la manière suivante : « b_n désigne le nombre d'additions effectuées lors du calcul de `fibo(n)` »

Q 2.4 Quelle est l'équation de récurrence de b_n ?

Exercice 3

On donne le code PYTHON commenté suivant :

```

def fibo_aux(n,i,cur,suiv):
    """ compute recursively the term of index n of the
    Fibonacci sequence
    assuming

    :param n: the index of the term
    :type n: int
    :param i: current index computed 0<=i<=n
    :type i: int
    :param cur: term of index i
    :type cur: int
    :param suiv: term of index i+1
    :type suiv: int
    :rtype: int
    :UC: all params have type int. n, i are non negative
    i is in [0,n]. raise AssertionError
    :Examples:

    >>> fibo_aux(0,0,0,1)
    0
    >>> fibo_aux(1,0,0,1)
    1
    >>> fibo_aux(6,0,0,1)
    8

```

```

>>> fibo_aux(6,2,1,2)
8
"""
assert type(n)==type(i)==type(cur)==type(suiv)==int
assert 0<=i<=n
if n==i:
    return cur
else:
    return fibo_aux(n,i+1,suiv,cur+suiv)

def fibo(n):
    """ compute the term of index n of the
    Fibonacci sequence using an auxiliary
    recursive function

    :param n: the index of the term
    :type n: int
    :rtype: int
    :UC: n is a non negative int. if not raise AssertionError
    :Examples:

    >>> fibo(0)
    0
    >>> fibo(1)
    1
    >>> fibo(5)
    5
    >>> fibo(6)
    8
    """
    assert type(n)==int and n>=0
    return fibo_aux(n,0,0,1)

```

Q 3.1 Lire le code !

La fonction `fibo` implante le calcul d'une suite définie par récurrence.

Q 3.2 Précisez laquelle ? (pas son nom mais son équation de récurrence !)

On définit la suite a de la manière suivante : « a_n désigne le nombre d'appels à la fonction `fibo_aux` effectués lors du calcul de `fibo(n)` »

Q 3.3 Quelle est l'équation de récurrence de a_n ? On définit la suite b de la manière suivante : « b_n désigne le nombre d'additions effectuées lors du calcul de `fibo(n)` »

Q 3.4 Quelle est l'équation de récurrence de b_n ?

Exercice 4

On reprend la célèbre suite de Heron vue en cours :

$$u_0 = 2$$

$$\forall n \in \mathbb{N} \quad u_{n+1} = \frac{\left(\frac{2}{u_n} + u_n\right)}{2}$$

et on définit la suite v par

$$\forall n \in \mathbb{N} \quad v_n = u_n - \sqrt{2}.$$

Q 4.1 Donnez une définition de v_n par récurrence.

Q 4.2 Démontrez par récurrence que pour tout entier n , le terme d'indice n de la suite v est positif ou nul.

Q 4.3 Démontrez par récurrence que pour tout entier n , le terme d'indice n de la suite v est inférieur ou égal à 1.

Q 4.4 Démontrez par récurrence que pour tout entier n , le terme d'indice n de la suite v est inférieur ou égal à $\frac{1}{2^n}$.

Exercice 5

On donne le code PYTHON non commenté suivant :

```
def f(x):  
    return 2.0*(x+1.0/(x*x))/3.0
```

```
def x(n):  
    if n == 0:  
        return 2  
    else:  
        return f(x(n-1))
```

Q 5.1 Lire le code!

La fonction x implante le calcul d'une suite définie par récurrence.

Q 5.2 Précisez laquelle?

On définit la suite a de la manière suivante : « a_n désigne le nombre d'appels à la fonction f effectués lors du calcul de $x(n)$ »

Q 5.3 Quelle est l'équation de récurrence de a_n ?