

ENSAE mardi 15 octobre 2013

Cette interrogation écrite compte pour 10 points ajoutés à l'ensemble des notes de la matière. Tous les documents sont autorisés. La durée est de 45 minutes. Vous devrez imprimer le résultat en n'omettant pas d'y ajouter votre nom et le numéro de l'énoncé qui vous aura été distribué.

1

1) Le triangle de Pascal sert à calculer les combinaisons, sa formule est :

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad (1)$$

On l'a implémenté comme suit :

```
pascal = { }
for i in range(1,100) :
    pascal[i,0] = 1
    pascal[i,i] = 1
    for j in range(0,i) :
        pascal [i,j] = pascal [i-1,j-1] + pascal[i-1,j]
print (pascal[5,4])
```

Le code contient une erreur (et une seule) car il produit l'exception suivante :

```
Traceback (most recent call last):
  File "interro_rapide_45_minutes_2013_12_1.py", line 10, in <module>
    pascal [i,j] = pascal [i-1,j-1] + pascal[i-1,j]
KeyError: (0, -1)
```

Corriger l'erreur d'indice sachant qu'elle se produit à l'une des trois lignes A,B,C? (2 points)

2) Ecrire une fonction qui calcule la somme des chiffres d'un entier positif. (4 points)

3) Un nombre est divisible par 11 lorsque la différence entre la somme des chiffres de rang pair et la somme des chiffres de rang impair est un multiple de 11. Ecrire une fonction qui détermine si un nombre est divisible par 11. (4 points)

Correction

1) L'erreur était à la ligne C. Si j commence à 0, on entre dans la seconde boucle lorsque i vaut 1. i - 1 vaut alors 0. Or les premières valeurs est dans le dictionnaire sont pour les indices (1,0) et (1,1). Il faut donc écrire `for j in range(0,i) :`

2) La réponse la plus courte consistait à utiliser une conversion en chaîne de caractères :

```
def somme_chiffre (i) :
    return sum ( [ int(c) for c in str(i) ] )

print (somme_chiffre(199))
```

Une réponse plus explicite que la première consistait à effectuer soi-même la somme des chiffres :

```
def somme_chiffre (i) :
    s = 0
    while i > 0 :
        s += i % 10
        i //= 10
    return s
```

3) Il fallait s'inspirer de la question précédente. Pour vérifier que la différence est divisible par 11, il est préférable de ne pas utiliser de division ni de modulo car c'est justement ce qu'on cherche à résoudre. On utilise la même fonction pour vérifier que la différence des deux sommes est divisible par 11 (on suppose donc que cette différence est inférieure au nombre initial).

```
def division_11 (i) :
    spair = 0
    simpair = 0
    pos = 1 # la première position est 1
    while i > 0 :
        if pos % 2 == 0 : spair += i % 10
        else : simpair += i % 10
        i //= 10
        pos += 1

    diff = abs(spair - simpair)

    if diff == 0 : return True
    elif diff < 11 : return False
    elif diff == 11 : return True
    else : return division_11(diff)

for i in [7,11,55,100,121,1001,3003,4000] :
    print (i,division_11(i))
```

ENSAE mardi 15 octobre 2013

Cette interrogation écrite compte pour 10 points ajoutés à l'ensemble des notes de la matière. Tous les documents sont autorisés. La durée est de 45 minutes. Vous devrez imprimer le résultat en n'omettant pas d'y ajouter votre nom et le numéro de l'énoncé qui vous aura été distribué.

2

1) On calcule la suite de Fibonacci comme suit :

```
n = 100
fibonacci = [ 0 ] * (n+1)          # A
fibonacci[0] = 1
for i in range(2,n) :             # B
    fibonacci[i] = fibonacci[i-1] + fibonacci[i-2]    # C
print (fibonacci) #affiche zéro
```

Malheureusement, le résultat affiché est nul. Corriger le problème sachant que l'erreur est située à une des trois lignes A,B,C. (2 points)

2) Ecrire une fonction qui retourne les diviseurs d'un nombre entier ? (5 points)

3) Ecrire une fonction qui détermine si un nombre entier est parfait ? (3 points) Un nombre entier n est parfait si la somme de ses diviseurs est égale à $2n$.

Correction

1) L'erreur est la ligne B. Le dernier indice calculé est $n-1$ et non n . Il suffit d'écrire `for i in range(2, n + 1) :` pour corriger le programme.

2) Il suffisait de vérifier que toutes les entiers compris entre 1 et $n/2$ divisait n .

```
def diviseur (i) :
    div = [ ]
    for d in range (1, i//2+1) :
        if i % d == 0 : div.append(d)
    return div

print (diviseur (11))
print (diviseur (512))
```

3) On utilise la fonction précédente :

```
def nombre_parfait (i) :
    div = diviseur(i)
    return sum(div) == 2*i

for i in range(115,125) :
    print (i, nombre_parfait(i), diviseur(i))
```