

DS d'informatique - CORRECTION

Durée : 2h

1. Écrire un programme qui, étant données deux bornes entières a et b, additionne les nombres multiples de 3 et de 5 compris entre ces bornes. Prendre par exemple a = 0, b = 32 ; le résultat devrait être alors 0 + 15 + 30 = 45. Modifier légèrement ce programme pour qu'il additionne les nombres multiples de 3 ou de 5 compris entre les bornes a et b. Avec les bornes 0 et 32, le résultat devrait donc être : 0 + 3 + 5 + 6 + 9 + 10 + 12 + 15 + 18 + 20 + 21 + 24 + 25 + 27 + 30 = 225.

```
• a=int(input('borne min'))
• b=int(input('borne max'))
• s=0
• c=0
• for i in range(a,b+1):
•     if i%3==0 and i%5==0:s=s+i
•     if i%3==0 or i%5==0:c=c+i
• print('la somme des multiples de 3 et 5 set',s)
• print('la somme des multiples de 3 ou 5 set',c)
```

2. Déterminer si une année (dont le millésime est introduit par l'utilisateur) est bissextile ou non. Une année A est bissextile si A est divisible par 4. Elle ne l'est cependant pas si A est un multiple de 100, à moins que A ne soit multiple de 400.

```
• a=int(input('entree annee'))
• if a%4==0:
•     if a%100==0 and a%400!=0:print('non bissextile')
•     else: print('bissextile')
• else: print('non bissextile')
```

3. Demander à l'utilisateur d'entrer trois longueurs a, b, c. À l'aide de ces trois longueurs, déterminer s'il est possible de construire un triangle. Déterminer ensuite si ce triangle est rectangle, isocèle, équilatéral ou quelconque. Attention : un triangle rectangle peut être isocèle.

```

def max(l):
    for i in range(len(l)):
        for j in range(i, len(l)):
            if l[j] < l[i]:
                aux = l[i]
                l[i] = l[j]
                l[j] = aux
    return l

l = []
a = float(input())
b = float(input())
c = float(input())
l.append(a)
l.append(b)
l.append(c)
l = max(l)

if l[2] < l[0] + l[1]:
    print('Triangle constructible')
    if (l[2])**2 == (l[0])**2 + (l[1])**2: print('le triangle est rectangle')
    if l[0] == l[1]:
        if l[0] == l[2]: print('le triangle est equilateral')
        else: print('le triangle est isocèle')
else: print('le triangle est non constructible')

```

4. Soit la suite (u_n) définie par la relation de récurrence $\begin{cases} u_{n+1} = \sqrt{5u_n - 4} \\ u_1 = 8 \end{cases}$.

a) Ecrire une fonction prenant pour paramètre N qui renvoie u_N .

b) Ecrire une fonction prenant pour paramètre p et qui renvoie $u_1 + u_2 + \dots + u_p$.

c) La suite (u_n) converge vers 4. Ecrire une fonction prenant pour paramètre *arrondi* et qui renvoie le plus petit entier n tel que $u_n - 4 \leq \text{arrondi}$.

```

from math import *

def terme(n):
    #reçoit le rang N et renvoie uN
    s = 8
    for i in range(1, n):
        s = sqrt(5*s - 4)
    return s

def somme(n):
    s = 0
    for i in range(1, n+1): s = s + terme(i)
    return s

def rang(arrondi):
    n = 1
    while terme(n) - 4 > arrondi: n = n + 1
    return n

```

5. Écrire une boucle de programme qui demande à l'utilisateur d'entrer des notes d'élèves. La boucle se terminera seulement si l'utilisateur entre une valeur négative. Avec les notes ainsi entrées, construire progressivement une liste.

Après chaque entrée d'une nouvelle note (et donc à chaque itération de la boucle), afficher le nombre de notes entrées, la note la plus élevée, la note la plus basse, la moyenne de toutes les notes.

A la fin de la saisie, faire afficher la liste triée des notes.

```
• l=[]
• s=0
• nbre_note=0
• note=float(input('entrez une nouvelle note positive, négative pour interrompre'))
• min=note
• max=note
• while note>=0:
•     l.append(note)
•     nbre_note=nbre_note+1
•     s=s+note
•     if note<min:min=note
•     if note>max:max=note
•     print('la moyenne est',s/nbre_note)
•     print('la note la plus basse est',min)
•     print('la meilleure note est',max)
•     note=float(input('entrez une nouvelle note positive, négative pour interrompre'))
• print('les notes rentrées étaient',l)
```