

TP 3

Q0 Quelles sont les six structures fondamentales des algorithmes ?

Binaire

Q1 Convertir 0x7FF en décimal

Q2 Convertir 1000 en binaire

Q3 Convertir 0x3CF0 en binaire

```
In [ ]: def fonctionMystere(n):  
        s=''  
        while n:  
            s=str(n%2)+s  
            n=n//2  
        return s
```

Q4 Que fait la fonction mystère ?

Q5 Décrire en français l'algorithme employé par cette fonction mystère.

Q6 Ecrire une fonction logbin qui donne le plus petit entier k tel que $2^k > n$ où n est le paramètre de la fonction.

Les variables P,Q,R sont des Booléens.

Q7 Recopier et compléter le tableau avec True ou False.

P	Q	R	P or not Q	not P or not R	not Q or R	(P or not Q) and (not P or not R) and (not Q or R)
False	False	False				
False	False	True				
False	True	False				
False	True	True				
True	False	False				
True	False	True				
True	True	False				
True	True	True				

Q9 Existe-t-il des valeurs booléennes pour a, b, c et d telles que la formule (not a or b or not c or d) and (a or b or not c or d) and (a or c or not d) soit vraie ?

Boucles

```
In [ ]: def somme(liste):  
        if liste==[]:  
            return 0  
        else:  
            return liste[0]+somme(liste[1:])  
  
        somme([3,5,7,9,11])
```

La fonction somme est *réursive*, elle s'appelle elle-même.

Q10 Que fait la fonction somme ?

Q11 Ecrire une fonction somme2 qui fait la même chose que somme mais avec une boucle, sans récursivité.

```
In [ ]: n=0  
        s=0  
        while s<100:  
            n=n+1  
            s=s+n
```

Q12 Que contient la variable n à la fin de la boucle ci-dessus ?

Q13 Ecrire une fonction qui prenne un seuil A en paramètre (p.ex 100) et qui renvoie le plus petit entier n tel que $1 + 2 + 3 + \dots + n \geq A$

```
In [ ]: liste=[]  
        for k in range(10):  
            liste.append(k**2)
```

Q14 Que contient la variable liste à la fin de la boucle ?

Q15 Que fait la *méthode* append ?

Motifs

```
In [ ]: for k in range(10):  
        print('*',end='')
```

L'option end='' permet que print ne passe pas à la ligne à chaque appel.

Q16 Ecrire un programme qui écrit une ligne de 80 *, puis une ligne de 80 /.

Q17 Ecrire une fonction ligne(n) qui permet d'écrire une ligne de n caractères *.

Q18 Ecrire un programme permettant d'afficher

On peut utiliser la fonction ligne.

Q19 Ecrire une fonction ligne2(n,carac) qui permet d'afficher n fois le caractère carac.

Par exemple carac(10, '/') devra afficher //////////.

Q20 Ecrire une fonction rectangle(m,n,carac) qui permet d'afficher un rectangle de m lignes et n colonnes du caractère carac. Par exemple l'affichage de la question 18 doit pouvoir être obtenu par rectangle(3,10, '*').

Les questions fondamentales des algorithmes

```
In [ ]: def DEFP(n):  
        liste=[]  
        p=2  
        while n:  
            if n%p==0:  
                liste.append(p)  
                n=n//p  
            else:  
                p=p+1  
        return liste
```

Q21 Que renvoie DEFP(1750) ?

Q22 Que fait la fonction DEFP ?

Q23 Comment peut-on être sûr que la fonction DEFP **termine** ?

Q24 Comment peut-on être sûr que la liste renvoyée ne contient que des nombres premiers, et que leur produit est le paramètre n ?

In []:

In []: