

NOM :**1. Test :**

Compléter la fonction suivante qui renvoie le prix d'une monture de lunettes vendu par un opticien qui fait la publicité suivante : "Si vous avez moins de 60 ans, recevez une remise d'un pourcentage égal à votre âge et si vous avez 60 ans ou plus, recevez une remise égale à votre âge".

Par exemple, un client de 28 ans aura 28% de remise et mamie qui a 74 ans aura 74€ de remise.

```

1 def prixmonture(x,n) :
2     """ Cette fonction renvoie le prix final d'une monture coûtant x euros après réduction éventuelle
       pour un client âgé de n ans """
3
4
5
6

```

Bernard a 59 ans. Il doit s'acheter une nouvelle monture. Expliquer ce que fait cette fonction Python.

```

1 def achat() :
2     """
3
4     x=60 # Prix minimum d'une monture dans ce magasin
5     while prixmonture(x,59)>prixmonture(x,60) :
6         x+=1
7     return x

```

Que doit faire Bernard si le prix de la monture choisie est inférieure à la valeur renournée par la fonction ?
Et s'il est supérieur ?

2. Suites récurrentes :

On considère la suite récurrente définie par $u_0 = 2$ et $\forall n \in \mathbb{N}$, $u_{n+1} = -\frac{2}{3}u_n^2 + \frac{u_n}{2} - 1$

On veut connaître pour un n donné la valeur de u_n ainsi que $S = \sum_{k=0}^n u_k$.

Ecrire une fonction Python qui remplit ces deux objectifs.

```

1 def recusomme(n) :
2     """ Cette fonction renvoie  $u_n$  et  $\sum_{k=0}^n u_k$  pour un entier n donné """
3
4
5
6
7
8     return u,s

```

3. Autour des listes de nombres :

Ecrire une fonction produitliste qui renvoie la valeur du produit des éléments d'une liste donnée :

```

1 def produitliste(L) :
2     """ Cette fonction calcule et renvoie le produit des éléments d'une liste L donnée """
3
4
5
6
7

```

Compléter la fonction suivante qui détermine le minimum d'une liste L donnée :

```

1 def minimumliste(L) :
2     """ Cette fonction détermine et renvoie le minimum d'une liste de nombres L donnée """
3     n=len(L) # taille de la liste L
4     mini=L[0] # minimum provisoire
5
6
7
8     return mini

```

4. **Entiers et factorions :** Expliquer ce que fait la fonction suivante :

```

1 def liste_entier(n) :
2     """
3
4     x=str(n)
5     p=len(x)
6     L=p*[0]
7     for k in range(p) :
8         L[k]=int(x[k])
9     return L

```

Que donne : liste_entier(145) ?

Ecrire la fonction factorielle qui étant donné un entier n renvoie $n!$, puis compléter les fonctions suivantes :

```

1 def factorielle(n) :
2     """ Cette fonction renvoie  $n!$  """
3
4
5
6

```

```

1 def factorion(n) :
2     """ Cette fonction teste si un nombre est un factorion, ie si ce nombre est égal à la somme des
3         factorielles des chiffres qui le compose. Elle renverra True dans ce cas, False sinon. """
4     L=liste_entier(n) # On transforme n en la liste de ses chiffres
5     S=0 # initialisation d'un somme
6     for c in L : # on parcourt L
7         S+=
8
9
10
11

```

Que donne factorion(125) ?

Et factorion(145) ?

```

1 def Listes_factorion(N) :
2     """ Cette fonction renvoie la liste des factorions inférieurs ou égaux à N. """
3     L=[ ] # initialisation de la liste vide
4
5
6
7     return L

```