

#EXERCICE 1

```
class Article:
    def __init__(self,ref,dsg,prix_a):
        self.ref=ref
        self.dsg=dsg
        self.prix_a=prix_a

    def __repr__(self):
        return '('+str(self.ref)+','+self.dsg+', '+str(self.prix_a)+')'

    def set_article(self,type_modif,modif,):
        if type_modif==1:
            if type(modif)!=int:
                raise ValueError('erreur de type')
            self.ref=modif

            elif type_modif==2:
                if type(modif)!=str:
                    raise ValueError('erreur de type')
                self.dsg=modif
            elif type_modif==3:
                if type(modif)!=float:
                    raise ValueError('erreur de type')
                self.prix_a=modif

    def prix_vente(self,P=20):
        return self.prix_a*(1+P/100)

class ArticleEnPromo(Article):
    def __init__(self,A,remise):
        Article.__init__(self,A.ref,A.dsg,A.prix_a)
        self.remise=remise
    def prix_ventePromo(self):
        return self.prix_vente()*(1-self.remise/100)

class Stock:
    def __init__(self):
        self.LS=[]

    def __repr__(self):
        ch=""
        for i in self.LS:
```

```
        ch=ch+str(i[0])+':'+' '+str(i[1])+'\n'  
    return ch
```

```
def recherche_article(self,ref):  
    for i in self.LS:  
        if i[0].ref==ref:  
            return i[0].dsg,i[0].prix_vente(),i[1]  
    return None
```

```
def ajouter_article(self,A,qt):  
    self.LS.append([A,qt])
```

```
def maj_qtstock(self,ref,qt,type_maj):  
    for i in self.LS:  
        if i[0].ref==ref:  
            if type_maj==1:  
                i[1]+=qt  
            else:  
                i[1]-=qt  
            break
```

```
def supprimer_Articles(self,ref):  
    for i in range(len(self.LS)):  
        if self.LS[i][0].ref==ref:  
            self.LS.pop(i)
```

```
def articles_prix(self,p1,p2):  
    L=[]  
    for i in self.LS:  
        pv=i[0].prix_vente()  
        if pv>=p1 and pv<=p2:  
            L.append(i[0])  
    return L
```

```
def articles_promo_qt(self,qt_min,remise):  
    L=[]  
    for i in self.LS:  
        if i[1]<qt_min:  
            x=ArticleSolde(i[0],remise)  
            L.append((x,x.prix_vente(),x.prix_venteSolde()))  
    return L
```

```
#main pour tester -
```

```
ref=int(input("donner la ref d'un article "))
```

```
A1=Article(1,"art1",200.)
```

```
A2=Article(2,"art2",300.)
```

```
A3=Article(3,"art3",98)
```

```
A4=Article(4,"art4",325)
```

```
A5=Article(5,"art5",67)
```

```
St1=Stock()
```

```
St1.ajouter_article(A1,25)
```

```
St1.ajouter_article(A2,30)
```

```
St1.ajouter_article(A3,40)
```

```
print(St1)
```

```
St2=Stock()
```

```
St2.ajouter_article(A1,250)
```

```
St2.ajouter_article(A2,300)
```

```
St2.ajouter_article(A5,400)
```

```
St3=Stock()
```

```
St3.ajouter_article(A4,16)
```

```
St3.ajouter_article(A2,30)
```

```
St3.ajouter_article(A3,20)
```

```
#création du dictionnaire StockE de 3 magasins
```

```
StockE={1:St1,2:St2,3:St3}
```

```
qt_T=0
```

```
for m in StockE:
```

```
    for i in StockE[m].LS:
```

```
        if i[0].ref==ref:
```

```
            print('Article dispo en magasin:', m,':qte=',i[1])
```

```
            qt_T+=i[1]
```

```
            break
```

```
print("qte totale=",qt_T)
```

