Correction Examen Programmation Python
Session Principale Janvier 2023
1ère année Génie Informatique

1) 2.25

```
def gen_livre():
    with open(chemin1, 'r') as f:  #0.5
        f.readline() #0.25
        data= reader(f, delimiter=",") #0.25
        yield from [{'isbn':l[0], 'titre':l[1],'genre': l[2],
                     'prix_actuel': eval(l[3])} for l in data] #1.25
```

2) 1.5

```
f= lambda : tuple(i for i in gen_livre()) #1.5
```

3) 3.75

```
def gen_ligneHistorique(isbn):
    with open(chemin2, 'r') as f: #0.5
        f.readline()#0.25
        for i in f: #0.25
            l=i.split(",") #0.25
            if l[0]==isbn: #0.5
                l1=l[1].split("/") #0.25
                l2=l1[0].split("-") #0.25
                yield [int(l2[0]), int(l2[1]), int(l2[2]), eval(l1[1])]#1.5
```

4) 2.5

```
def verifier(isbn):
    g= gen_ligneHistorique(isbn)#0.25
    try: #0.5
        next(g) #0.5
    except Exception as e:  #0.5
        raise ValueError("Article n'existe pas")#0.5
    return True #0.25
```

5) 1.75

```
def chercher_prix(isbn):
    l=[el for el in gen_ligneHistorique(isbn)]#0.5
    assert l!=[] #0.5
    return l[l.index(max(l))][-1] #0.75
```

6) 2.5

```
def ajouter_livre(mode, **liv):
    with open(chemin1, mode=mode, newline="\n") as f: #0.25
        wr= DictWriter(f, fieldnames=list(liv), delimiter=",") #0.5
        if mode=="w" or mode=="w+": wr.writeheader() #0.25
        if mode=="a" or mode=="a+": #0.25
            for j in gen_livre(): #0.25
                assert liv['isbn']!=j['isbn'] #0.75
        wr.writerow(liv) #0.25
```

7) 2.25

```
def prix_actuel():
     for el in t_livres: #0.25
            try: #0.25
                verifier(el['isbn']) #0.5
                el['prix_actuel']= chercher_prix(el['isbn']) #0.5
            except Exception as e: #0.25
                el['prix_actuel']= -1 #0.5
```

8) 3.75

```
def modifier_fichier():
    global t_livres #0.25
    l=sorted(t_livres,key=lambda d:d['prix_actuel'],reverse=True)#0.75
    #Supprimer les livres ayant "prix_actuel=-1"
    for d in l[::-1]: #0.25 (parcours inverse)
      if d['prix_actuel']==-1: #0.25
        l.remove(d) #0.25
      else: break #0.25
    t_livres=tuple(l) #0.25 (conversion liste--> tuple)
    ajouter_livre(mode='w+', **t_livres[0])#0.5 (appel correct)
    try: #0.25 (try+except)
        for el in t_livres[1:]: #0.25
            ajouter_livre(mode='a+', **el)#0.5
    except Exception as e:
        print(e)
```

##Partie optionnelle##

9) 0.75

```
def gen_livre_p(f):
  for d in t_livres:#0.25
    if f(d):  #0.25
      yield d #0.25
```

10) 2.25

```
f1= lambda  : "/". join(list(map(lambda d:d['isbn'] ,
                gen_livre_p(lambda d: d['genre']=='Romanesque'
                        and d['prix_actuel']>15 )
                                )))
```

#0.25 (lambda)
#0.25 ("/". Join)
#0.25 (list(…))
#0.25 (map(…))
#0.25 (lambda d:d['isbn'])
#0.25 (gen_livre_p(.....))
#0.75 (lambda d: d['genre']=='Romanesque' and d['prix_actuel']>15)