

# PTSI2022\_TP2\_corrige

October 17, 2022

## 0.1 Corrigé TP Bases programmation 2

### Q1.

```
[13]: entree = input().split()
      sortie = ""          # nouvelle chaine
      for l in entree:    # pour chaque lettre,
          if l != " ":   # qui n'est pas un espace
              sortie = sortie + l # on l'ajoute
      print(sortie)
```

Un moment viendra où tu changeras de place.  
Unmomentviendraoùtuchangerasdeplace.

### Q2.

```
[19]: entree = input()
      c = 0
      for l in entree:
          if l == 'e':
              c+=1
      print("Il y a",c,"e dans cette phrase")
```

Un moment viendra où tu changeras de place.  
Il y a 5 e dans cette phrase

### Q3.

```
[16]: entree = input()
      sortie = ""
      for l in entree:    # chaque lettre
          sortie = l + sortie # passe devant
      print(sortie)
```

Un moment viendra où tu changeras de place.  
.ecalp ed saregnahc ut ùo ardneiv tmemom nU

### Q4.

```
[18]: entree = input()
      c = 0
```

```

sortie = ""
for l in entree:
    if c%2 == 0:
        sortie = sortie + l
    c += 1
print(sortie)

```

Le lycée Livet  
L y é i e

### Q5.

```

[29]: # on supprime les espaces avec string.replace(" ", "")
entree = "engage le jeu que je le gagne".replace(" ", "")

```

```

[30]: palindrome = True # hypothèse de travail
for k in range(len(entree)):
    if entree[k] != entree[len(entree)-k-1] :
        palindrome = False # si deux lettres ne correspondent pas
print(str(palindrome))

```

True

### Q6.

```

[1]: e = [[2014, "masculin", "belge", "Dries Feremans", 110.42],
[2018, "masculin", "allemand", "Tom Philipp Reinhardt", 102.68],
[2012, "masculin", "finlandais", "Ere Karjalainen", 101.46],
[2012, "masculin", "tchèque", "Tomáš Benda", 97.34],
[2012, "masculin", "sud-africain", "Jeremy Gallop", 94.67],
[2017, "féminine", "autrichienne", "Ivonne Wiener", 67.58],
[2012, "féminine", "tchèque", "Tereza Kopicová", 60.24],
[2012, "féminine", "tchèque", "Kateřina Reehová", 54.68],
[2006, "féminine", "britannique", "Jan Singleton", 53.52],
[2006, "féminine", "finlandaise", "Eija Laakso", 50.83]]

```

```

[3]: for i in e:
    print("En {} l'athlète {} {} {} a lancé son telephone à {} mètres."
        ↪format(i[0],i[1],i[2],i[3],i[4]))

```

En 2014 l'athlète masculin belge Dries Feremans a lancé son telephone à 110.42 mètres.

En 2018 l'athlète masculin allemand Tom Philipp Reinhardt a lancé son telephone à 102.68 mètres.

En 2012 l'athlète masculin finlandais Ere Karjalainen a lancé son telephone à 101.46 mètres.

En 2012 l'athlète masculin tchèque Tomáš Benda a lancé son telephone à 97.34 mètres.

En 2012 l'athlète masculin sud-africain Jeremy Gallop a lancé son telephone à

94.67 mètres.

En 2017 l'athlète féminine autrichienne Ivonne Wiener a lancé son telephone à 67.58 mètres.

En 2012 l'athlète féminine tchèque Tereza Kopicová a lancé son telephone à 60.24 mètres.

En 2012 l'athlète féminine tchèque Kateřina Reehová a lancé son telephone à 54.68 mètres.

En 2006 l'athlète féminine britannique Jan Singleton a lancé son telephone à 53.52 mètres.

En 2006 l'athlète féminine finlandaise Eija Laakso a lancé son telephone à 50.83 mètres.

**Q7.**

```
[35]: entree = input().split(" ")
      ref = entree[0][0]    # initiale
      tauto = True
      for mot in entree:
          if mot[0] != ref:
              tauto = False
      print(tauto)
```

mazarin, ministre malade, méditait même moribond malicieusement mille maltôtes.  
True

**Q8.**

```
[29]: def entiers(n):
      L=[]
      for k in range(n):
          L.append(k)
      return L
```

```
[33]: entiers(10), [k for k in range(10)]
```

```
[33]: ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

**Q9.**

```
[3]: def pairs(n):
      L = []
      entier = 0
      while entier <= n:
          L.append(entier)
          entier += 2
      return L
```

```
[4]: pairs(26), [i for i in range(0,26+1,2)]
```

```
[4]: ([0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26],  
      [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26])
```

### Q10.

```
[36]: def max2(x,y):  
       if x >= y:  
           return x  
       else:  
           return y
```

```
[37]: max2(-13,12)
```

```
[37]: 12
```

### Q11.

```
[38]: def moyenne(L):  
       somme = 0  
       for x in L:  
           somme += x  
       return somme/len(L)
```

```
[73]: moyenne([k**2 for k in range(10)])
```

```
[73]: 28.5
```

### Q12.

```
[39]: def somme(n):  
       s = 0  
       for k in range(n+1):  
           s = s + k  
       return s  
  
       def sommeCube(n):  
           s = 0  
           for k in range(n+1):  
               s = s + k**3  
           return s
```

```
[21]: sommeCube(10), (somme(10))**2
```

```
[21]: (3025, 3025)
```

### Q13.

```
[42]: import math
def suite(n):
    u = 1/2
    for k in range(n):
        u = math.sqrt(1-u)
    return u
```

```
[43]: suite(5)
```

```
[43]: 0.6572500918350371
```

#### Q14.

```
[44]: def somme_liste(L):
s = 0
for k in range(len(L)):
    s = s + L[k]
return s
```

```
[45]: somme_liste([k for k in range(100)])
```

```
[45]: 4950
```

#### Q15.

```
[46]: def rectangle(L,H):
ch= "+" + L*"- " + "\n"
for k in range(H):
    ch = ch + "| " + L*" " + "\n"
ch= ch + "+" + L*"- " + "\n"
print(ch)
```

```
[47]: rectangle(8,2)
```

```
+-----+
|       |
|       |
+-----+
```

#### Q16.

```
[8]: def bissextile(annee):
    if (annee%4==0 and annee%100 != 0) or annee%400==0:
        return True
    return False
```

```
[9]: bissextile(2020),bissextile(2021),bissextile(1900),bissextile(2000)
```

[9]: (True, False, False, True)

### Q17.

```
[58]: def asterix(chaine):
      nch=""
      for lettre in chaine:
          nch = nch + lettre + "*"
      return nch[:len(nch)-1] #on enlève la dernière asterisque
```

```
[59]: asterix("texte")
```

[59]: 't\*e\*x\*t\*e'

### Q18.

```
[60]: def presence_multiple(L,n):
      liste = []
      for k in range(len(L)):
          if L[k] % n ==0:
              liste.append(k)
      return liste
```

```
[61]: presence_multiple([k for k in range(14)],3)
```

[61]: [0, 3, 6, 9, 12]

### Q19.

```
[63]: def presence_caract(chaine,c):
      L = []
      for k in range(len(chaine)):
          if chaine[k]==c:
              L.append(k)
      return L
```

```
[65]: presence_caract("Un jour viendra où on changera de place.", "a")
```

[65]: [14, 24, 29, 36]

### Q20.

```
[68]: def inversion(ch):
      nouvelle = ""
      for k in range(len(ch)):
          nouvelle = ch[k] + nouvelle # ici, on inverse l'ordre
      return nouvelle
```

```

#OU
#def inversion(ch):
#    nouvelle = ""
#    for k in range(len(ch)):
#        nouvelle = nouvelle + ch[len(ch)-k-1] # on parcourt la chaîne à
#        ↪ l'envers
#    return nouvelle

```

```
[69]: inversion("Un jour viendra où on changera de place.")
```

```
[69]: '.ecalp ed aregnahc no ùo ardneiv ruoj nU'
```

### Q21.

```
[70]: def trinome(a,b,c):
    delta = b**2-4*a*c
    if delta == 0:
        return -b/(2*a)
    elif delta >0:
        return (-b + delta**0.5)/(2*a),(-b - delta**0.5)/(2*a)
    else :
        return (-b + 1j*(-delta)**0.5)/(2*a),(-b - 1j*(-delta)**0.5)/(2*a)

```

```
[72]: trinome(1,2,1),trinome(1,2,-3),trinome(1,2,3)
```

```
[72]: (-1.0, (1.0, -3.0), ((-1+1.4142135623730951j), (-1-1.4142135623730951j)))
```

```
[ ]:
```