

```
In [1]: def CreePileVide():  
        return []
```

```
In [2]: def TestePileVide(P):  
        return P==[]
```

```
In [3]: def Empile(p,x):  
        p.append(x)  
  
        def DepileRenvoie(p):  
            return p.pop()
```

```
In [4]: P=CreePileVide()
```

```
In [5]: Empile(P,2)
```

```
In [6]: Empile(P,8)
```

```
In [7]: P
```

```
Out[7]: [2, 8]
```

```
In [8]: DepileRenvoie(P)
```

```
Out[8]: 8
```

```
In [9]: P
```

```
Out[9]: [2]
```

```
In [10]: def hauteur(p):  
        compteur=0  
        while TestePileVide(p)==False:  
            compteur+=1  
            DepileRenvoie(p)  
        return compteur
```

```
In [11]: def popsecond(p):  
        x=DepileRenvoie(p)  
        y=DepileRenvoie(p)  
        Empile(p,x)  
        return p
```

```
In [12]: def popfond(p):  
        pp=CreePileVide()  
        while TestePileVide(p)==False:  
            Empile(pp,DepileRenvoie(p))  
        DepileRenvoie(pp)  
        while TestePileVide(pp)==False:  
            Empile(p,DepileRenvoie(pp))  
        return p
```

```
In [13]: def copier_coller(P):
        Q=CreePileVide()
        R=CreePileVide()
        # on dépile Les éléments de P dans Q
        while TestePileVide(P)==False:
            x=DepileRenvoie(P)
            Empile(Q,x)
        #On dépile Les élément de Q dans R et P
        while TestePileVide(Q)==False:
            x=DepileRenvoie(Q)
            Empile(P,x)# ne pas oublier cette ligne sionon P est une pile vide
            Empile(R,x)
        return R
```

```
In [14]: copier_coller(P)
```

```
Out[14]: [2]
```

```
In [15]: Empile(P,8)
```

```
In [16]: Empile(P,1)
```

```
In [17]: P
```

```
Out[17]: [2, 8, 1]
```

```
In [18]: def inverser(P):
        cP=copier_coller(P)# copie de la liste de départ
        Q=CreePileVide()
        while TestePileVide(cP)==False:
            x=DepileRenvoie(cP)
            Empile(Q,x)
        return Q
```

```
In [19]: P,inverser(P),P
```

```
Out[19]: ([2, 8, 1], [1, 8, 2], [2, 8, 1])
```

```
In [ ]:
```