

- 1- Dessiner une feuille de copie un ABR en insérant à partir d'un arbre vide, dans l'ordre, les clés :
50 – 25 – 75 – 12 – 30 - 60

Le code incomplet ci-dessous permet d'implémenter l'ABR ci-dessus :

```
class Arbre() :
    def __init__(self, info , fg = None , fd = None) :
        self.info = str(info)
        self.fg = fg
        self.fd = fd

    def insererElement(self, val):
        if val < int(self.info) :
            if self.fg == None :
                new = Arbre(val)
                self.fg = new
                return new
            else : self.fg.insererElement(val)
        else :
            if self.fd == None :
                new = Arbre(val)
                self.fd = new
                return new
            else : self.fd.insererElement(val)

    def rechercherElement(self, val) :
        if val == int(self.info) : return True
        else :
            if val < int(self.info) :
                if self.fg == None :
                    return False
                else :
                    return self.fg.rechercherElement(val)
            else :
                if self.fd == None :
                    return False
                else :
                    return self.fd.rechercherElement(val)

    def max(self) :
        if self.fd != None :
            return self.fd.max()
        else :
            return self.info

    def estABR(self) :
        if self.fg != None and self.fd != None :
            if int(self.info) > int(self.fg.info) and int(self.info) < int(self.fd.info) :
                return self.fg.estABR() and self.fd.estABR()
            else : return False
        elif self.fg != None :
            if int(self.info) > int(self.fg.info) : return self.fg.estABR()
            else : return False
        elif self.fd != None :
            if int(self.info) < int(self.fd.info) : return self.fd.estABR()
            else : return False
        else :
            return True
```

```
# Main
A = Arbre(50)
A.insererElement(25)
A.insererElement(75)
A.insererElement(12)
A.insererElement(30)
A.insererElement(60)

reponse = A.rechercherElement(25)
print(f"\n25 est-il dans l'arbre ? : {reponse}")

maxi = A.max()
print(f"\nLa valeur maximale contenue dans l'arbre est : {maxi}")

reponse = A.estABR()
print(f"\nL'arbre est-il un ABR ? : {reponse}")
```

L'exécution de ce code donne :

```
25 est-il dans l'arbre ? : True

La valeur maximale contenue dans l'arbre est : 75

L'arbre est-il un ABR ? : True
```

- 2- Compléter sur feuille de copie, le code de la méthode *insererElement()*. Ne compléter que la partie correspondante à la zone grisée.
- 3- Compléter sur feuille de copie, le programme principal afin de pouvoir créer l'ABR de la question 1 précédente.
- 4- Compléter le code de la méthode *rechercherElement()*
- 5- Compléter le code de la méthode *max()* et de la partie du programme principal qui appelle cette méthode.
- 6- Compléter le code de la méthode *estABR()* et de la partie du programme principal qui appelle cette méthode.