

```

class Nodo:
    def __init__(self, value=None, izq=None, der=None):
        self.value = value
        self.izq = izq
        self.der = der

    def __str__(self):
        return self.value

class aBinarios:
    def __init__(self):
        self.raiz = None
    def agregar(self, elemento):

        if self.raiz == None:
            self.raiz = elemento
        else:
            aux = self.raiz
            padre = None
            while aux != None:
                padre = aux
                if int(elemento.value) >= int(aux.value):
                    aux = aux.der
                else:
                    aux = aux.izq
            if int(elemento.value) >= int(padre.value):
                padre.der = elemento
            else:
                padre.izq = elemento

    def preorden(self, elemento):
        if elemento != None:
            print(elemento)
            self.preorden(elemento.izq)
            self.preorden(elemento.der)

    def postorden(self, elemento):
        if elemento != None:
            self.postorden(elemento.izq)
            self.postorden(elemento.der)
            print(elemento)

    def inorden(self, elemento):
        if elemento != None:
            self.inorden(elemento.izq)
            print(elemento)
            self.inorden(elemento.der)

    def getRaiz(self):
        return self.raiz

if __name__ == "__main__":

```

```
ab = aBinarios()
while(True):
    print("Arboles_Binarios\n"+
          "1. Agregar\n"+
          "2. Preorden\n"+
          "3. Postorden\n"+
          "4. Inorden\n"+
          "5. Salir")

    num = input("ingrese la opcion: ")

    if num == "1":
        value = input("Ingrese el valor: ")
        nod =Nodo(value)
        ab.agregar(nod)

    elif num == "2":
        print("Preorden")
        ab.preorden(ab.getRaiz())
    elif num == "3":
        print("postorden")
        ab.postorden(ab.getRaiz())
    elif num == "4":
        print("inorden")
        ab.inorden(ab.getRaiz())
    elif num == "5":
        exit()
```