Assignement 2:

Test case 1:

```java
class GFG {

    static int ternarySearch(int a, int b, int key, int ar[])
    {
     if (b >= a) {


       int mid1 = a + (b − a) / 3;
       int mid2 = b − (b − a) / 3;

       if (ar[mid1] == key) {
        return mid1;
       }
       if (ar[mid2] == key) {
        return mid2;
       }


       if (key < ar[mid1]) {

         return ternarySearch(a, mid1 − 1, key, ar);
       }
       else if (key > ar[mid2]) {

         return ternarySearch(mid2 + 1, b, key, ar);
       }
       else {

         return ternarySearch(mid1 + 1, mid2 − 1, key, ar);
       }
     }

      return −1;
    }


//test case 1:

    public static void main(String args[])
    {
     int a, b, p, key;
```

```
        int ar[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };


    a= 0;

    b = 9;

    key = 8;
    p = ternarySearch(a, b, key, ar);
    System.out.println("Index of " + key + " is " + p);
  }
 }
```

Output:
harshavaidhyam@Harshas-MacBook-Pro assignement part quiz 2 % cd
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design/Assignment\ week\
2/assignement\ part\ quiz\ 2 ; /usr/bin/env /Library/Java/JavaVirtualMachines/temurin-
17.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInE
xceptionMessages -cp /Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design/Assignment\
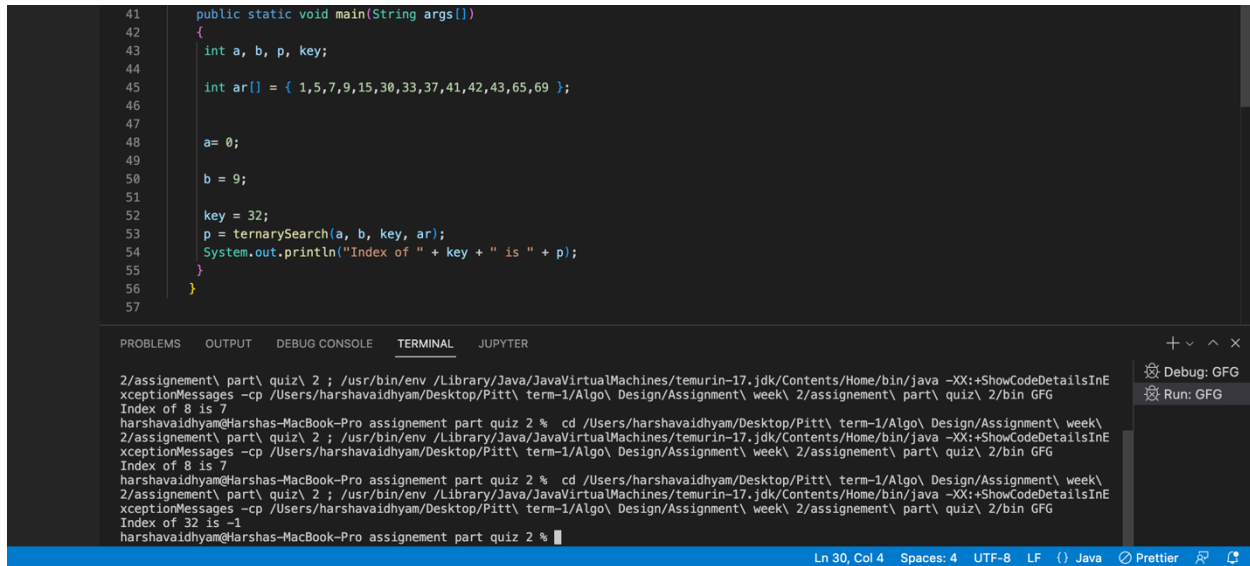week\ 2/assignement\ part\ quiz\ 2/bin GFG
Index of 8 is 7

Test case 2:

```
class GFG {


    static int ternarySearch(int a, int b, int key, int ar[])
    {
     if (b >= a) {


      int mid1 = a + (b − a) / 3;
      int mid2 = b − (b − a) / 3;

      if (ar[mid1] == key) {
       return mid1;
      }
      if (ar[mid2] == key) {
       return mid2;
      }
```

```java
            if (key < ar[mid1]) {

                return ternarySearch(a, mid1 - 1, key, ar);
            }
            else if (key > ar[mid2]) {

                return ternarySearch(mid2 + 1, b, key, ar);
            }
            else {

                return ternarySearch(mid1 + 1, mid2 - 1, key, ar);
            }
        }

        return -1;
    }


//test case 2:

    public static void main(String args[])
    {
        int a, b, p, key;

        int ar[] = { 1,5,7,9,15,30,33,37,41,42,43,65,69 };


        a= 0;

        b = 9;

        key = 32;
        p = ternarySearch(a, b, key, ar);
        System.out.println("Index of " + key + " is " + p);
    }
}
```

Output:

Index of 32 is -1

Order Notation:
$O(\log_3 n)$

Screenshots: