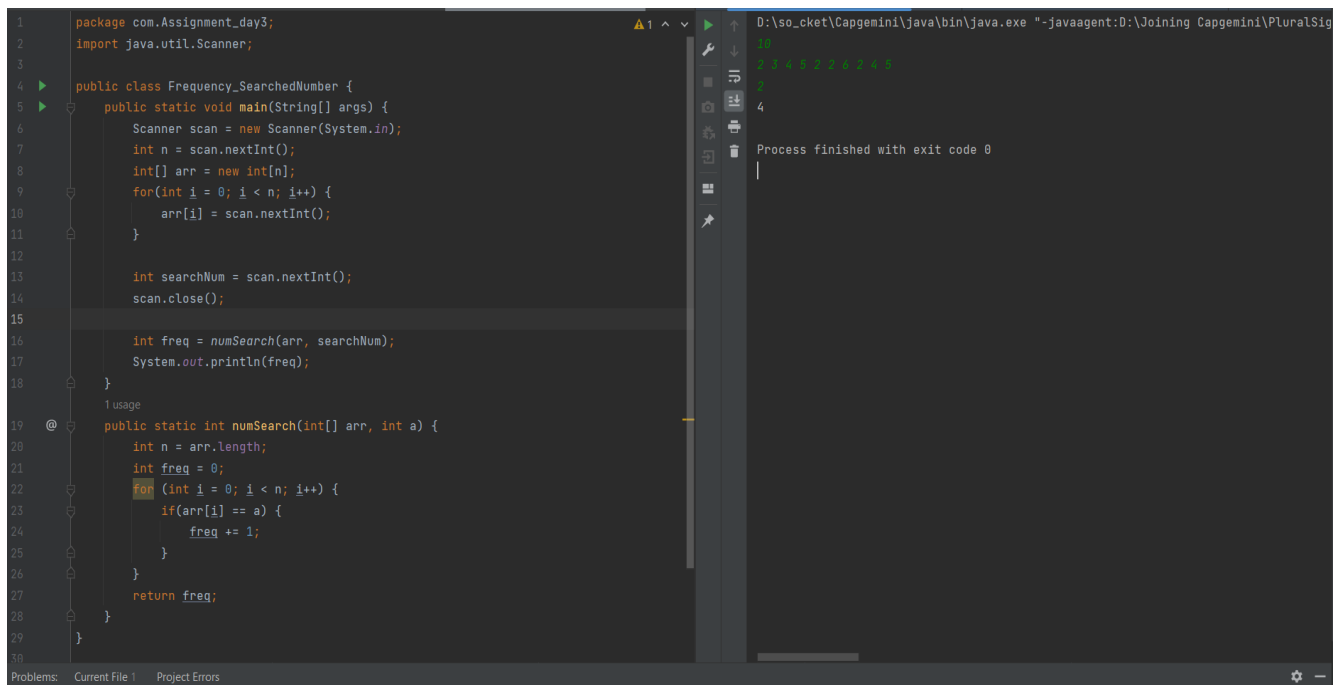


## Day 3 – Assignment

**Pratik K Kamble**  
**Employee ID : 46263548**  
**12 – 09 – 2022**

**Q1:** Ten numbers are entered from the keyboard into an array. The number to be searched is entered through the keyboard by the user. Write a function which should get the array as input and a number to be searched. This function should return the frequency of a searched number in provided list of numbers.

**Solution –**



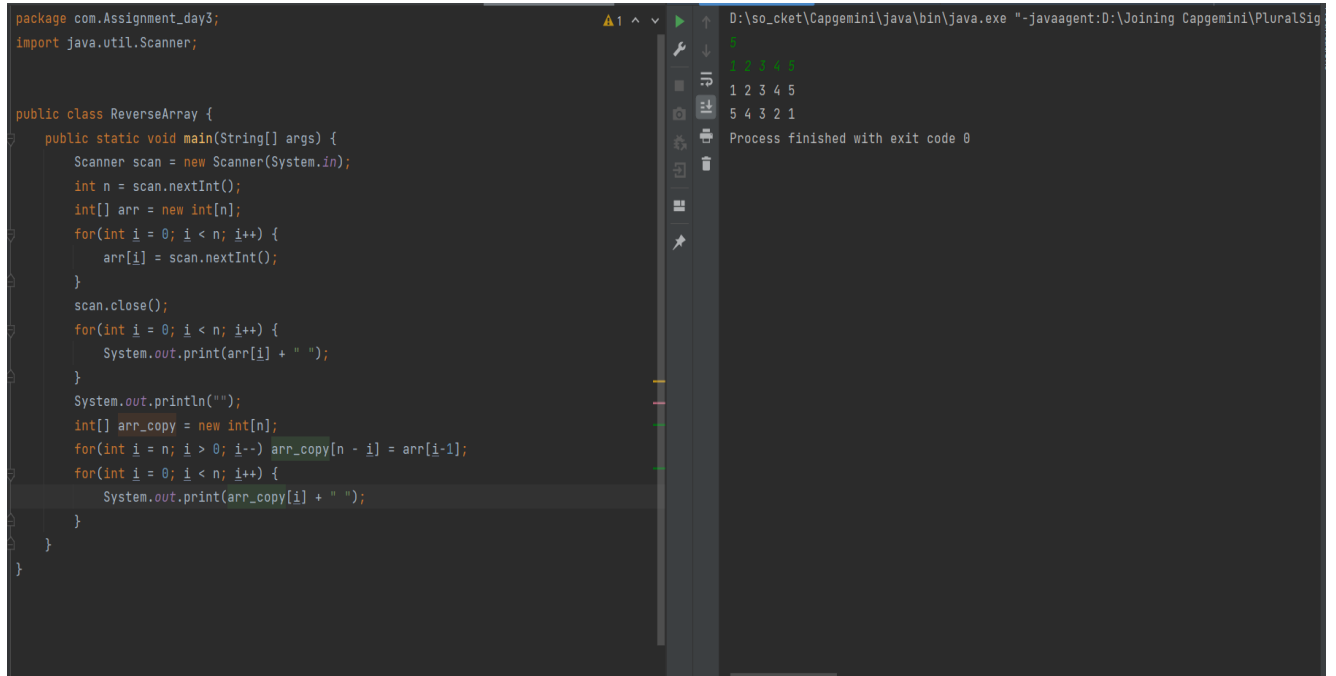
```
1 package com.Assignment_day3;
2 import java.util.Scanner;
3
4 public class Frequency_SearchedNumber {
5     public static void main(String[] args) {
6         Scanner scan = new Scanner(System.in);
7         int n = scan.nextInt();
8         int[] arr = new int[n];
9         for(int i = 0; i < n; i++) {
10             arr[i] = scan.nextInt();
11         }
12
13         int searchNum = scan.nextInt();
14         scan.close();
15
16         int freq = numSearch(arr, searchNum);
17         System.out.println(freq);
18     }
19
20     public static int numSearch(int[] arr, int a) {
21         int n = arr.length;
22         int freq = 0;
23         for (int i = 0; i < n; i++) {
24             if(arr[i] == a) {
25                 freq += 1;
26             }
27         }
28         return freq;
29     }
30 }
```

The screenshot shows an IDE with the following components:

- Editor:** Contains the Java code for the `Frequency_SearchedNumber` class. The code includes a `main` method that reads 10 integers into an array and a search number, and a static `numSearch` method that returns the frequency of the search number in the array.
- Run Console:** Shows the output of the program. It displays the input numbers: `10 1 2 3 4 5 6 7 8 9 10` and the search number: `4`. The final output is `4`, indicating the frequency of the number 4 in the array.
- Status Bar:** Indicates that the process finished with exit code 0.

**Q2:** Write a program to copy the contents of one array into another array in the reverse order, without using any inbuilt method.

**Solution –**



The screenshot shows a Java IDE with a code editor on the left and a console on the right. The code in the editor is as follows:

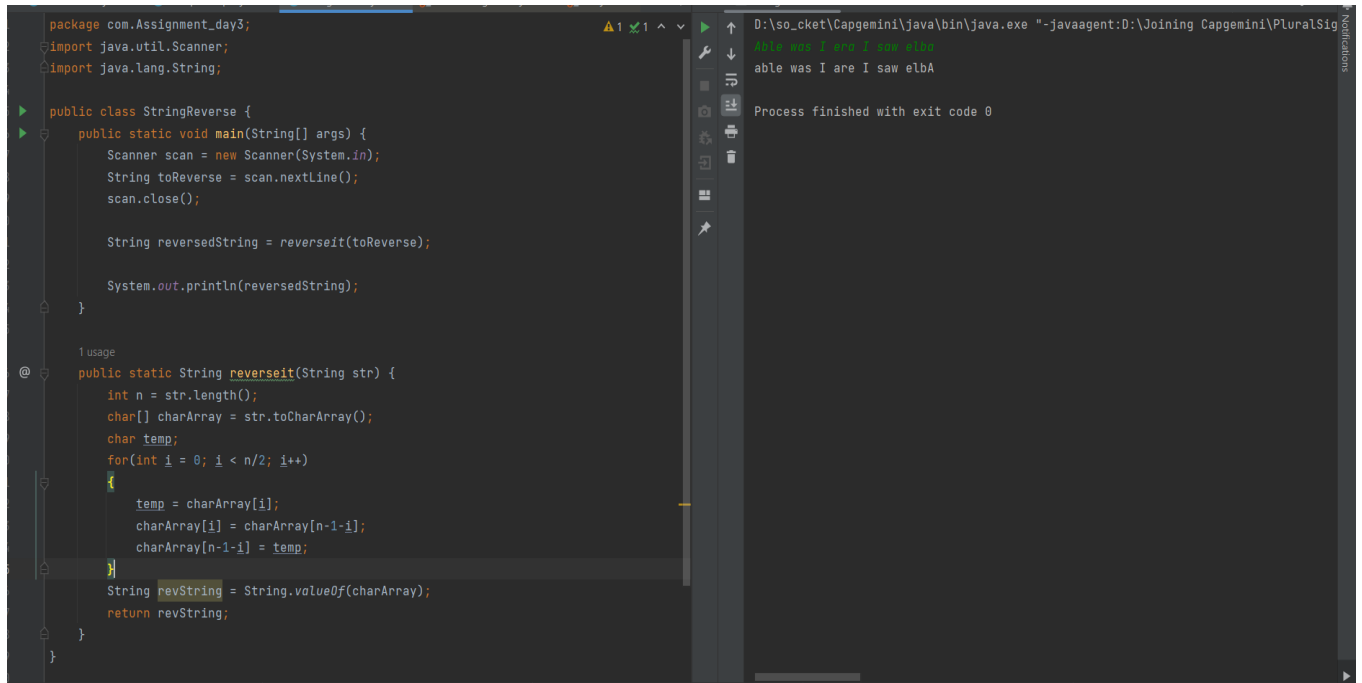
```
package com.Assignment_day3;
import java.util.Scanner;

public class ReverseArray {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        int[] arr = new int[n];
        for(int i = 0; i < n; i++) {
            arr[i] = scan.nextInt();
        }
        scan.close();
        for(int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
        int[] arr_copy = new int[n];
        for(int i = n; i > 0; i--) arr_copy[n - i] = arr[i-1];
        for(int i = 0; i < n; i++) {
            System.out.print(arr_copy[i] + " ");
        }
    }
}
```

The console on the right shows the output of the program. It displays the input array "1 2 3 4 5" and the reversed array "5 4 3 2 1". The process finished with exit code 0.

**Q3:** Write a function `reverseit()`, that reverse a String. Use a for loop that swaps the first and last characters, then second and next-to-last characters and so on. The string should be passed to `reverseit()` method as an argument and it should return the reverse string. Write a program to exercise `reverseit()`. The program should get the string from the user. Call `reverseit()` and print the output. Check the program with the following String "Able was I era I saw elba."

### Solution –



```
package com.Assignment_day3;
import java.util.Scanner;
import java.lang.String;

public class StringReverse {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String toReverse = scan.nextLine();
        scan.close();

        String reversedString = reverseit(toReverse);

        System.out.println(reversedString);
    }

    1 usage
    @ public static String reverseit(String str) {
        int n = str.length();
        char[] charArray = str.toCharArray();
        char temp;
        for(int i = 0; i < n/2; i++)
        {
            temp = charArray[i];
            charArray[i] = charArray[n-1-i];
            charArray[n-1-i] = temp;
        }
        String revString = String.valueOf(charArray);
        return revString;
    }
}
```

D:\soCKET\Cappgemini\java\bin\java.exe "-javaagent:D:\Joining Cappgemini\PluralSig  
able was I era I saw elbA  
able was I era I saw elbA  
Process finished with exit code 0

Q4: Create a 5 story car parking system simulation, where ground floor will be having the capacity to park 5 cars, 2nd story will have the capacity to park 4 cars, 3rd story will have the capacity to park 3 cars, 4th story will have the capacity to park 2 cars and finally 5th story will have the capacity to park 1 car.

ClassName	Car
Fields	regNo, ownerName, tokenNo
Methods	Generate all getter and setter and parameterized constructor
ClassName	CarParking
Methods	showAvailability() : this will show the available space
	parkCar(Car car): this will park the car on available place
	getCar(Car car): this will get the car for owner based on provided tokenNumber. And make the same place available for next parking
ClassName	CarParkingDemo
Method Name	main(String[] args): this will check all the operations of the CarParking.

## Solution –

```

public class CarParkingDemo {
    public static void main(String[] args) {
        CarParking carParking = new CarParking();
        carParking.showAvailability();

        Car car_1 = new Car( regNo: 1, ownerName: "M3", tokenNo: 51);
        Car car_2 = new Car( regNo: 2, ownerName: "AMG", tokenNo: 56);
        Car car_3 = new Car( regNo: 3, ownerName: "R8", tokenNo: 61);

        carParking.parkCar(car_1);
        carParking.parkCar(car_2);
        carParking.showAvailability();
        carParking.getCar(car_1);
        carParking.showAvailability();
    }
}

class Car{
    int regNo;
    String ownerName;
    int tokenNo;

    Car(int regNo, String ownerName, int tokenNo){
        this.regNo = regNo;
        this.ownerName = ownerName;
        this.tokenNo = tokenNo;
    }
}

```

Output:

```

D:\socket\Cappgemini\java\bin\java.exe --javaagent:D:\Joinir
Floor 1: 0 0 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 5 4 3 2 1
M3 Parked in Floor: 1
AMG Parked in Floor: 1
Floor 1: 51 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 3 4 3 2 1
Parking is free now on floor: 1
Floor 1: 0 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 4 4 3 2 1
Process finished with exit code 0

```

```
public int getRegNo(){
    return regNo;
}

public void setRegNo(int regNo){
    this.regNo = regNo;
}

public String getOwnerName(){
    return ownerName;
}

public void setOwnerName(String ownerName){
    this.ownerName = ownerName;
}

public int getTokenNo() {
    return tokenNo;
}

public void setTokenNo(int tokenNo){
    this.tokenNo = tokenNo;
}

}

2 usages
class CarParking{
    7 usages
    int [][] parkingSpaces;
    6 usages
    final static int[] floorCapacity = new int[]{5,4,3,2,1};
}
```

D:\so\_cket\Capgemini\java\bin\java.exe "-javaagent:D:\Joinir  
Floor 1: 0 0 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 5 4 3 2 1  
M3 Parked in Floor: 1  
AMG Parked in Floor: 1  
Floor 1: 51 56 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 3 4 3 2 1  
Parking is free now on floor: 1  
Floor 1: 0 56 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 4 4 3 2 1  
Process finished with exit code 0

```
//Initializing Parking Spaces
1 usage
CarParking(){
    this.parkingSpaces = new int[][]{{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
}

3 usages
void showAvailability(){
    int[] capacity = new int[]{0,0,0,0,0};
    for (int i = 0; i < floorCapacity.length; i++)
    {
        System.out.print("Floor " + (i+1) + ": ");
        for(int j = 0; j < floorCapacity[i]; j++)
        {
            if(parkingSpaces[i][j] == 0){
                capacity[i] += 1;
            }
            System.out.print(parkingSpaces[i][j] + " ");
        }
        System.out.println();
    }
    System.out.print("Available Parkings: ");
    for(int i = 0; i < capacity.length; i++){
        System.out.print(capacity[i] + " ");
    }
    System.out.println();
}

}

D:\so_cket\Capgemini\java\bin\java.exe "-javaagent:D:\Joinir  
Floor 1: 0 0 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 5 4 3 2 1  
M3 Parked in Floor: 1  
AMG Parked in Floor: 1  
Floor 1: 51 56 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 3 4 3 2 1  
Parking is free now on floor: 1  
Floor 1: 0 56 0 0 0  
Floor 2: 0 0 0 0  
Floor 3: 0 0 0  
Floor 4: 0 0  
Floor 5: 0  
Available Parkings: 4 4 3 2 1  
Process finished with exit code 0
```

```
2 usages
void parkCar(Car car){
    boolean isParked = false;
    int floor = 0;
    for(int i = 0; i < floorCapacity.length; i++) {
        for(int j = 0; j < floorCapacity[i]; j++) {
            if (parkingSpaces[i][j] == 0) {
                isParked = true;
                parkingSpaces[i][j] += car.tokenNo;
                floor = i+1;
                break;
            }
        }
        if(isParked)
        {
            break;
        }
    }
    if(isParked){
        System.out.println(car.ownerName + " Parked in Floor: " + floor);
    }
    else {
        System.out.println("Free Space for Parking.");
    }
}

D:\soCKET\Capgemini\java\bin\java.exe "-javaagent:D:\Joini
Floor 1: 0 0 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 5 4 3 2 1
M3 Parked in Floor: 1
AM6 Parked in Floor: 1
Floor 1: 51 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 3 4 3 2 1
Parking is free now on floor: 1
Floor 1: 0 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 4 4 3 2 1
Process finished with exit code 0
```

```
1 usage
void getCar(Car car) {
    boolean isfree = false;
    int floor = 0;
    for(int i = 0; i < floorCapacity.length; i++) {
        for(int j = 0; j < floorCapacity[i]; j++) {
            if(parkingSpaces[i][j] == car.tokenNo) {
                isfree = true;
                parkingSpaces[i][j] = 0;
                floor = i+1;
                break;
            }
        }
        if(isfree){
            break;
        }
    }
    if(isfree){
        System.out.println("Parking is free now on floor: " + floor);
    }
}

D:\soCKET\Capgemini\java\bin\java.exe "-javaagent:D:\Joini
Floor 1: 0 0 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 5 4 3 2 1
M3 Parked in Floor: 1
AM6 Parked in Floor: 1
Floor 1: 51 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 3 4 3 2 1
Parking is free now on floor: 1
Floor 1: 0 56 0 0 0
Floor 2: 0 0 0 0
Floor 3: 0 0 0
Floor 4: 0 0
Floor 5: 0
Available Parkings: 4 4 3 2 1
Process finished with exit code 0
```

**Q5:** You need to create your own ArrayUtil class that will be providing you multiple utility methods that will be helping you out to perform operations on Any array.

Note: Try to attempt this exercise using TDD. (It is optional)

Class Name	ArrayUtil
Method	searchUsingBinarySearch(int[] arr, int key): this will search the specified int array for the specified key using binary search algorithm.
	searchUsingBinarySearch(float[] arr, float key): this will search the specified float array for the specified key using binary search algorithm.
	searchUsingBinarySearch(Object[] arr, object key): this will search the specified object array for the specified key using binary search algorithm.
	static int[] arrayCopy(int[] original, int newLength): Copies the specified array, newLength is the increased length of the new array.
	static void sort(int[] arr, int fromIndex, int toIndex): Sorts the specified range of specified array of objects into ascending order according to the natural ordering of its elements.
Important Note	You need to work on float and object variation for sort and arrayCopy method as well.
Restrictions	You are not suppose to use the inbuilt methods for this exercise.
ClassName	CheckArrayUtil
Method name	main(String[] args):This method is used to test the ArrayUtil class operations.

## Solution –

```
package com.Assignment_day3;

import org.jetbrains.annotations.NotNull;

import java.util.Arrays;

class Object{
    int value;
    public Object(int value) {
        this.value = value;
    }
}

public class ArrayUtil {
    int BinarySearch(int[] arr, int l, int r, int key) {

        if(r >= l) {
            int mid = l + (r - l) / 2;

            if (arr[mid] == key) {
                return mid;
            }

            if (arr[mid] > key) {
                return BinarySearch(arr, l, mid - 1, key);
            }

            return BinarySearch(arr, mid + 1, r, key);
        }

        return -1;
    }

    int searchUsingBinarySearch(int[] arr, int key){
        int l = 0;
        int r = arr.length-1;
        return BinarySearch(arr, l, r, key);
    }

    int BinarySearch(float[] arr, int l, int r, float key) {

        if(r >= l) {
            int mid = l + (r - l) / 2;

            if (arr[mid] == key) {
                return mid;
            }

            if (arr[mid] > key) {
                return BinarySearch(arr, l, mid - 1, key);
            }

            return BinarySearch(arr, mid + 1, r, key);
        }
    }
}
```



```

        return -1;
    }

    int searchUsingBinarySearch(float[] arr, float key){
        int l = 0;
        int r = arr.length-1;
        return BinarySearch(arr, l, r, key);
    }

    int BinarySearch(Object[] arr, int l, int r, int key) {

        if(r >= l) {
            int mid = l + (r - l) / 2;

            if (arr[mid].value == key) {
                return mid;
            }

            if (arr[mid].value > key) {
                return BinarySearch(arr, l, mid - 1, key);
            }

            return BinarySearch(arr, mid + 1, r, key);
        }

        return -1;
    }

    int searchUsingBinarySearch(Object[] arr, int key){
        int l = 0;
        int r = arr.length-1;
        return BinarySearch(arr, l, r, key);
    }

    void sort(int @NotNull [] arr, int fromIndex, int toIndex) {
        int i, key, j;
        for(i = fromIndex+1; i < toIndex; i++)
        {
            key = arr[i];
            j = i - 1;
            while(j >= 0 && arr[j] > key) {
                arr[j+1] = arr[j];
                j--;
            }
            arr[j+1] = key;
        }
        System.out.println("Sorted Integer Array: " + Arrays.toString(arr));
    }

    void sort(float @NotNull [] arr, int fromIndex, int toIndex) {
        int i, j;
        float key;
        for (i = fromIndex+1; i < toIndex; i++) {
            key = arr[i];
            j = i - 1;
            while (j >= 0 && arr[j] > key) {

```

```

        arr[j + 1] = arr[j];
        j--;
    }
    arr[j + 1] = key;
}
System.out.println("Sorted Float Array: " + Arrays.toString(arr));
}

void sort(Object @NotNull [] arr, int fromIndex, int toIndex){
//    int i, j;
//    Object key = new Object(3);
//    for (i = fromIndex+1; i < toIndex; i++) {
//        key.value = arr[i].value;
//        j = i - 1;
//        while (j >= 0 && arr[j].value > key.value) {
//            arr[j + 1] = arr[j];
//            j--;
//        }
//        arr[j + 1].value = key.value;
//    }
    int min;
    for(int i = fromIndex; i < toIndex; i++) {
        int minValue = arr[i].value;

        for(int j = i+1; j < toIndex; j++){
            if(arr[j].value < arr[i].value) {
                int temp = arr[j].value;
                arr[j].value = arr[i].value;
                arr[i].value = temp;
            }
        }
    }
    CheckArrayUtil arrayfun = new CheckArrayUtil();
    System.out.print("Sorted Object Array: ");
    arrayfun.printarr(arr);
}

int[] arrayCopy(int[] original, int newLength) {
    int[] newArr = new int[newLength];
    for (int i = 0; i < original.length; i++)
    {
        newArr[i] = original[i];
    }
    return newArr;
}

float[] arrayCopy(float[] original, int newLength) {
    float[] newArr = new float[newLength];
    for (int i = 0; i < original.length; i++)
    {
        newArr[i] = original[i];
    }
    return newArr;
}

Object[] arrayCopy(Object[] original, int newLength) {
    Object[] newArr = new Object[newLength];

```

```

        int len = Math.min(original.length, newLength);
        for (int i = 0; i < len; i++)
        {
            newArr[i] = original[i];
        }
        return newArr;
    }
}

class CheckArrayUtil{

    public void printarr(int[] arr){
        for(int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }

    public void printarr(float[] arr){
        for(int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }

    public void printarr(Object[] arr){

        for(int i = 0; i < arr.length; i++) {
            if (arr[i] == null){
                System.out.print(" null ");
            } else {
                System.out.print(arr[i].value + " ");
            }
        }
    }

    public static void main(String[] args) {
        int[] arr = new int[]{4, 6, 1, 10, 5};
        float[] arr2 = new float[]{8.1f, 5.1f, 10.1f, 11.1f, 4.1f};
        Object[] arr3 = new Object[]{new Object(78), new Object(15), new
Object(56), new Object(46), new Object(9)};

        CheckArrayUtil arrayfun = new CheckArrayUtil();

        System.out.print("Integer array is: ");
        arrayfun.printarr(arr);
        System.out.println("");
        System.out.print("Float array is: ");
        arrayfun.printarr(arr2);
        System.out.println("");
        System.out.print("Object array is: ");
        arrayfun.printarr(arr3);
        System.out.println("");
        System.out.println("");

        ArrayUtil funarray = new ArrayUtil();
        System.out.println("The index of given number in integer array is: "
+ funarray.searchUsingBinarySearch(arr, 10));
    }
}

```

```

        System.out.println("The index of given number in float array is: " +
funarray.searchUsingBinarySearch(arr2, 10.1f));
        System.out.println("The index of given number in Object array is: " +
funarray.searchUsingBinarySearch(arr3, 15));
        System.out.println("");

        ArrayUtil func = new ArrayUtil();
        System.out.println("Copied new integer Array is: " +
Arrays.toString(func.arrayCopy(arr, 7)));
        System.out.println("Copied new Float Array is: " +
Arrays.toString(func.arrayCopy(arr2, 7)));
        System.out.print("Copied new Object Array is: ");
        Object[] obj = funarray.arrayCopy(arr3, 7);
        arrayfun.printarr(obj);
        System.out.println("");
        System.out.println("");

        funarray.sort(arr, 0, arr.length);
        funarray.sort(arr2, 0, arr.length);
        funarray.sort(arr3, 0, arr.length);
        System.out.println("");

    }
}

```

```

D:\soCKET\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capgemini
Integer array is: 4 6 1 10 5
Float array is: 8.1 5.1 10.1 11.1 4.1
Object array is: 78 15 56 46 9

The index of given number in integer array is: 3
The index of given number in float array is: 2
The index of given number in Object array is: 1

Copied new integer Array is: [4, 6, 1, 10, 5, 0, 0]
Copied new Float Array is: [8.1, 5.1, 10.1, 11.1, 4.1, 0.0, 0.0]
Copied new Object Array is: 78 15 56 46 9 null null

Sorted Integer Array: [1, 4, 5, 6, 10]
Sorted Float Array: [4.1, 5.1, 8.1, 10.1, 11.1]
Sorted Object Array: 9 15 46 56 78

Process finished with exit code 0

```