**Day 3 – Assignment**

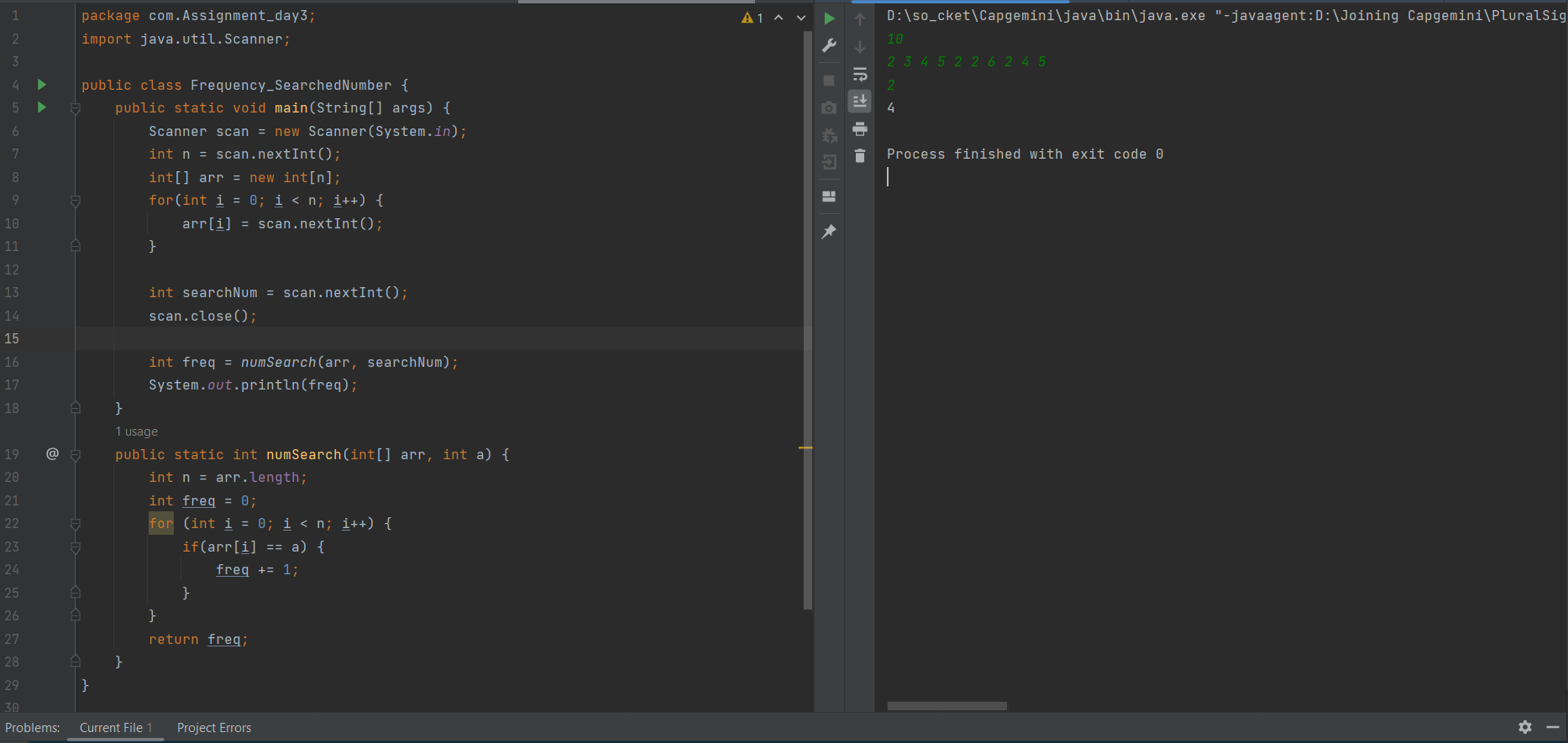
**Pratik K Kamble**

**Employee ID : 46263548**

**12 – 09 – 2022**

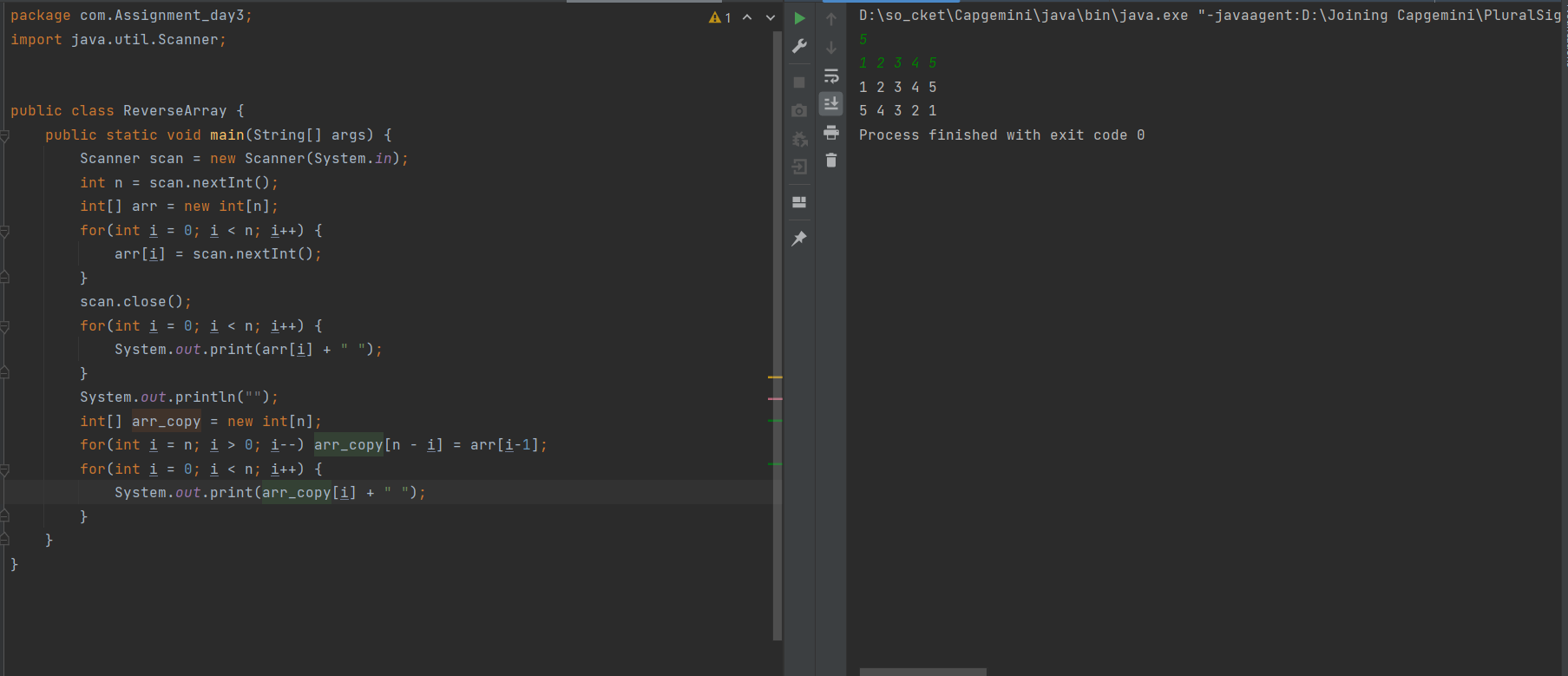
**Q1:** Ten numbers are entered from the keyboard into an array. The number to be searched is entered through the keyboard by the user. Write a function which should get the array as input and a number to be searched. This function should return the frequency of a searched number in provided list of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

**Solution –**

****

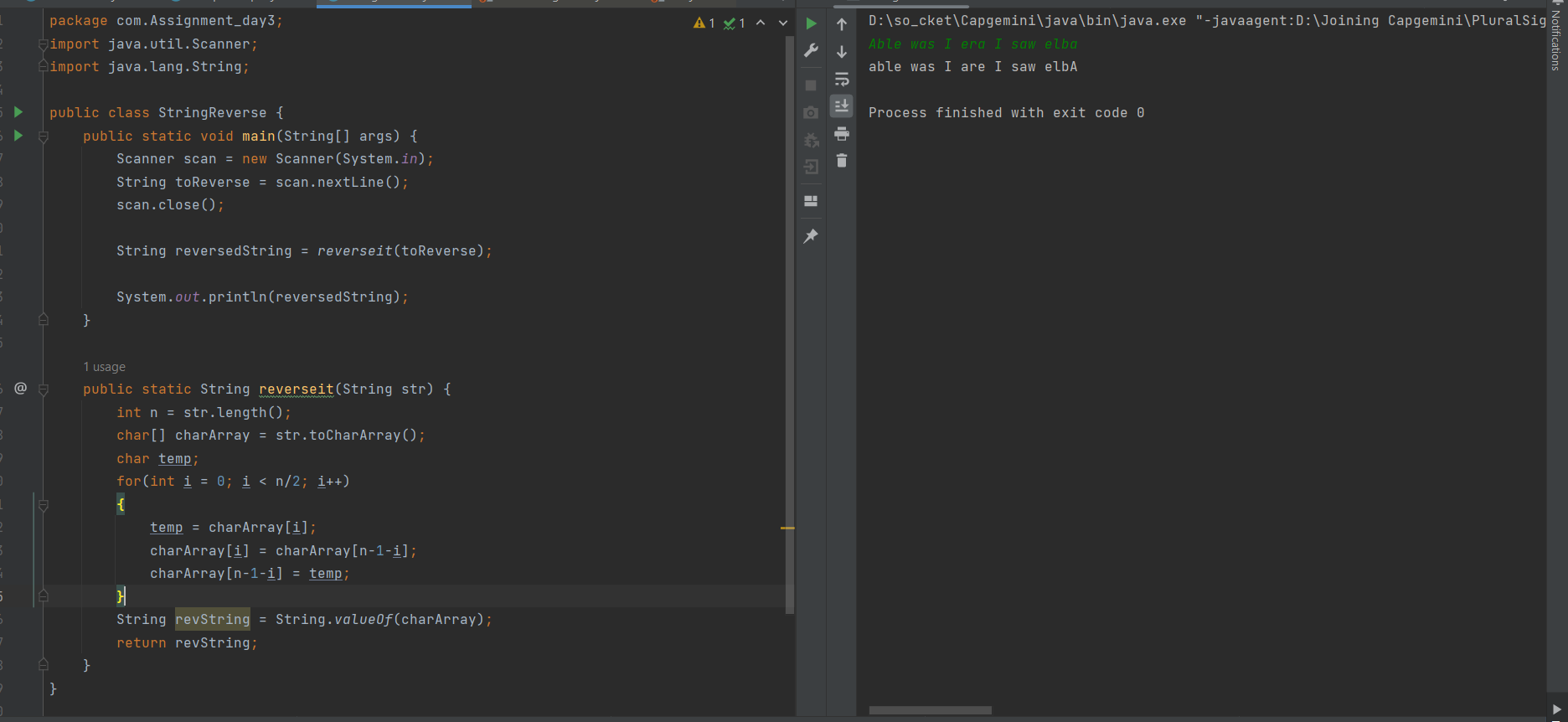
**Q2:** Write a program to copy the contents of one array into another array in the reverse order, without using any inbuilt method.

**Solution –**

****

**Q3:** Write a function revereseIf(), 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 revereseIt() 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.”.

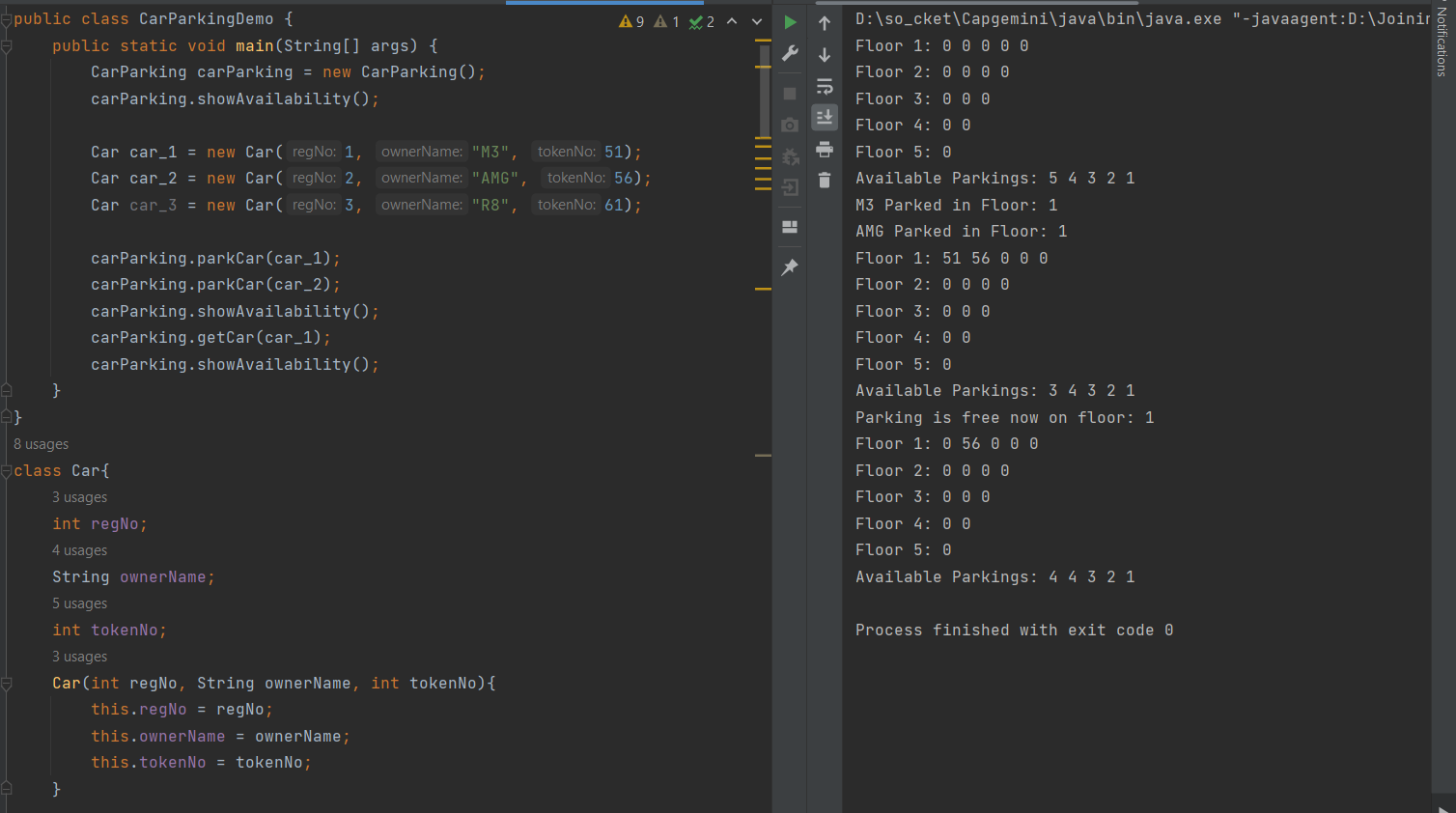
**Solution –**

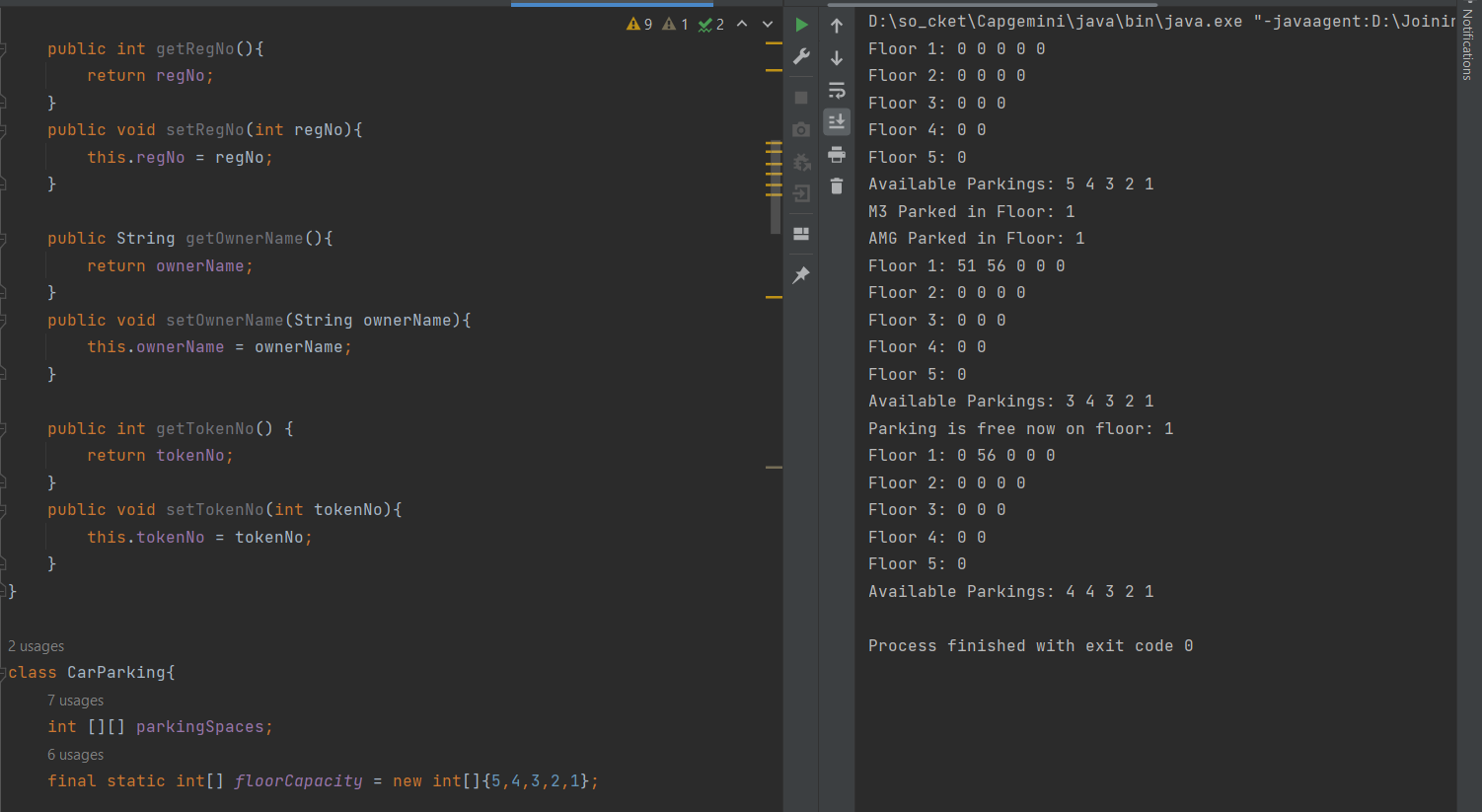
****

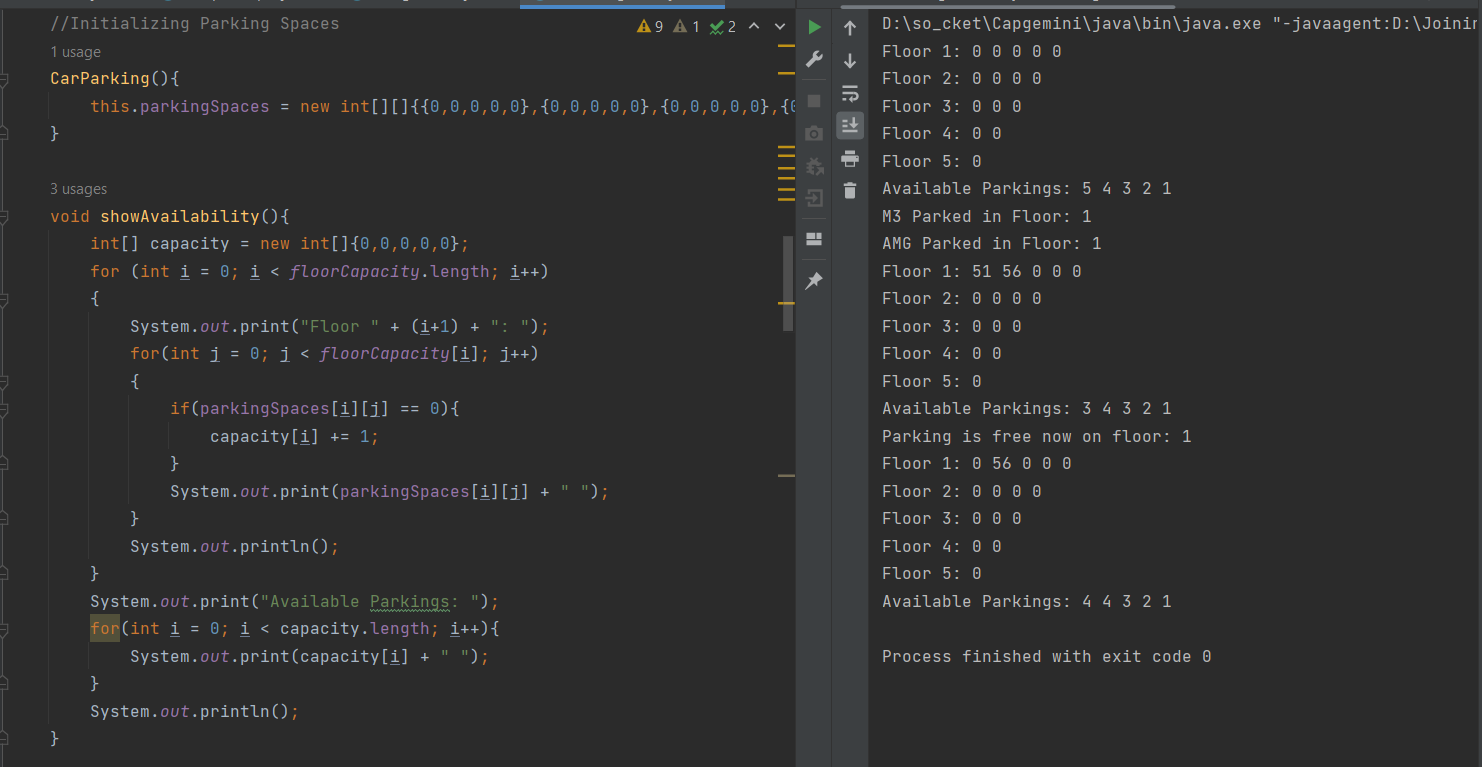
Q4: Create a 5 story car parking system simulation, where ground floor will be having the capacity to park 5 cars, 2nd story will have the capacity to park 4 cars, 3rd story will have the capacity to park 3 cars, 4th story will have the capacity to park 2 cars and finally 5th story will have the capacity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

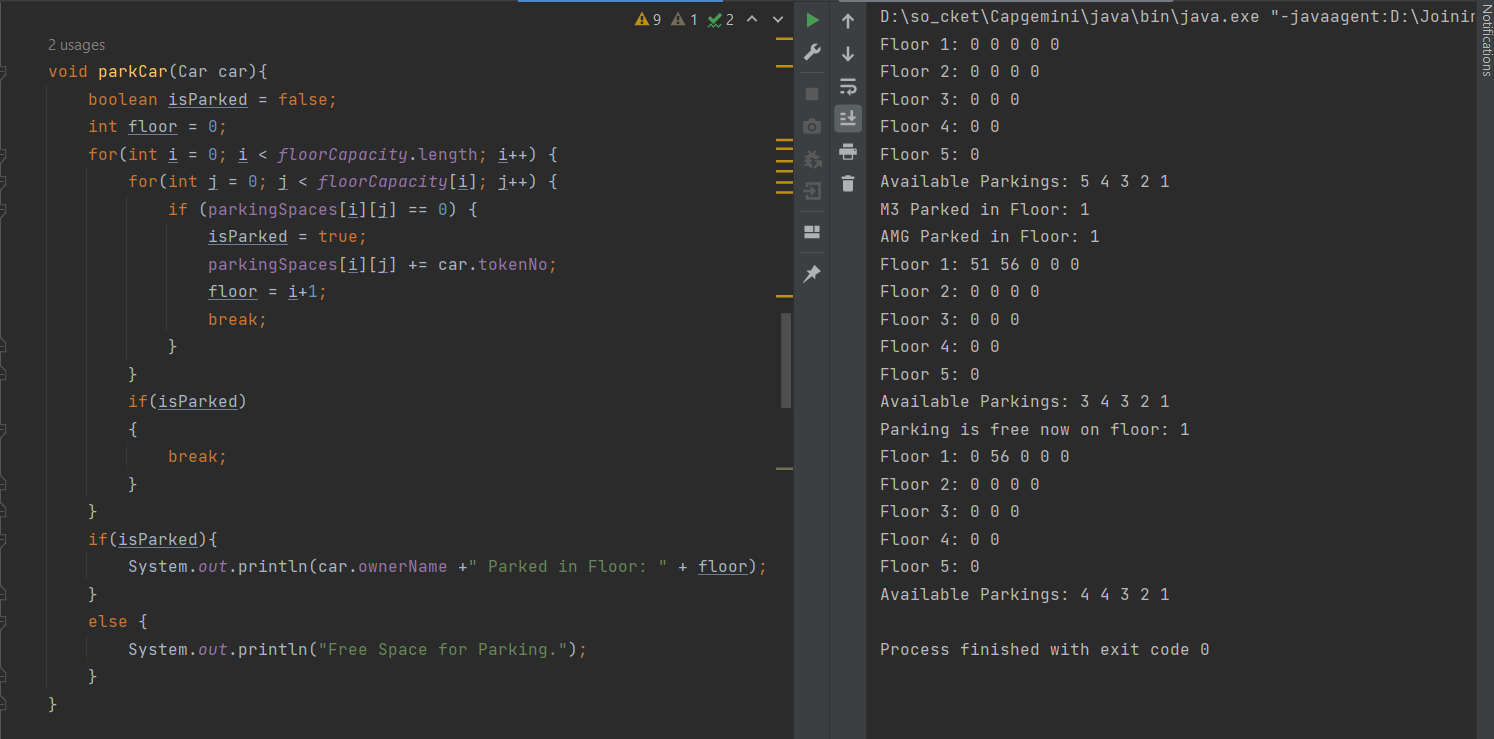
|  |  |
| --- | --- |
| ClassName | Car |
| Fields | regNo, ownerName, tokenNo |
| Methods | Generate all getter and setter and parameterized constructor |
| ClassName | CarParking |
| Methods | showAvailability() : this will show the available space |
|  | parkCar(Car car): this will park the car on 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. |

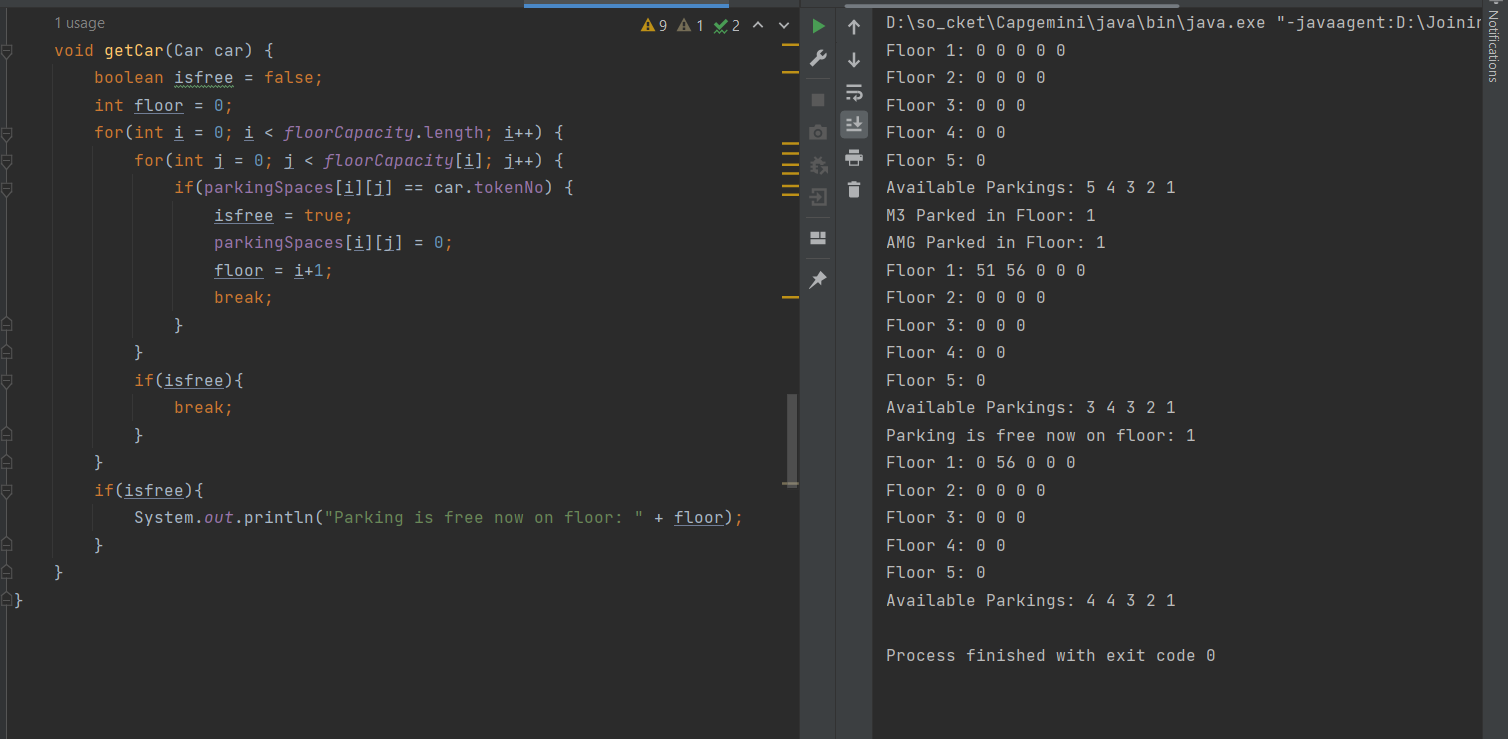
**Solution –**











**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, newLenth 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 - 1) / 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 - 1) / 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 - 1) / 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("");  
  
  
  
 }  
}

