ASSIGNMENT NO.6

1. Declare a single-dimensional array of 5 integers inside the `main` method. Traverse the array to print the default values. Then accept records from the user and print the updated values of the array.

```
package Assignment_6;
import java.util.Arrays;
import java.util.Scanner;
public class Q1 {
public static void main(String[] args) {
int arr[]=new int[5];
Scanner sc=new Scanner(System.in);
System.out.println(Arrays.toString(arr));
//int i=sc.nextInt();
for(int a=0; a<arr.length; a++) {
arr[a]=sc.nextInt();
}
System.out.println("Updated values are:"+Arrays.toString(arr));
sc.close();
}
}
```

Output:

```
[0, 0, 0, 0, 0]

4
2
4
3
3
Updated values are:[4, 2, 4, 3, 3]
```

2. Declare a single-dimensional array of 5 integers inside the `main` method. Define a method named `acceptRecord` to get input from the terminal into the array and another method named `printRecord` to print the state of the array to the terminal.

```
<terminated> Q9 [Java Application] C:\Program Files\Ja
Enter 5 integers:
Enter integer:1
3
Enter integer:2
4
Enter integer:3
2
Enter integer:4
9
Enter integer:5
6
Array contents:
Element 0:3
Element 1:4
Element 2:2
Element 3:9
Element 4:6
```

```
package Assignment_6;
```

ASSIGNMENT NO.6

```
import java.util.Scanner;

class ArrayExample {

public static void acceptRecord(int[] arr) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter 5 integers:");

for (int i = 0; i < arr.length; ++i) {

System.out.println("Enter integer:" + (i + 1));

arr[i] = sc.nextInt();

//sc.close();

}

}

public static void printRecord(int[] arr) {

System.out.println("Array contents:");

for(int i=0; i<arr.length; ++i) {

System.out.println("Element "+i+":"+arr[i]);

}

}

}

public class Q2 {

public static void main(String[] args) {

int[] numbers=new int[5];

ArrayExample.acceptRecord(numbers);

ArrayExample.printRecord(numbers);

}

}
```

3. Write a program to find the maximum and minimum values in a single-dimensional array of integers.

```
package Assignment_6;

import java.util.Scanner;

class MaxMinArray{

public static void acceptRecord(int[] arr) {

Scanner sc= new Scanner(System.in);

System.out.println("Enter 5 integers:");

for(int i=0; i<arr.length; ++i) {

arr[i]=sc.nextInt();

}

}

public static void printRecord(int[] arr) {

System.out.println("Array contents:");

for(int i=0; i<arr.length; i++) {

System.out.println( arr[i]);

}

}

public static int findMin(int[] arr) {

int min=arr[0];

for(int i=1; i<arr.length; ++i) {

min=Math.min(min, arr[i]);

}

return min;
```

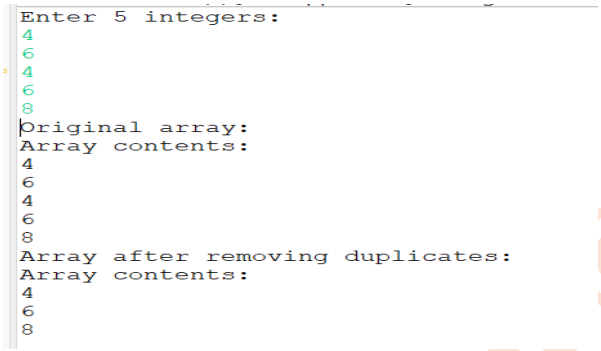
```
}
public static int findMax(int[] arr) {
int max=arr[0];
for(int i=1; i<arr.length; ++i) {
max=Math.max(max, arr[i]);
}
return max;
}
}
public class Q3 {
public static void main(String[] args) {
int arr[]=new int[5];
MaxMinArray.acceptRecord(arr);
MaxMinArray.printRecord(arr);
int min=MaxMinArray.findMin(arr);
int max=MaxMinArray.findMax(arr);
System.out.println("Minimum number is:"+min);
System.out.println("Maximum number is:"+max);
}
}
```

```
Enter 5 integers:
7
9
4
6
3
Array contents:
7
9
4
6
3
Minimum number is:3
Maximum number is:9
```

4. Write a program to remove duplicate elements from a single-dimensional array of integers.

```
package Assignment_6;
import java.util.Scanner;
class RemoveDuplicates {
// Method to get input from the terminal into the array
public static void acceptRecord(int[] arr) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter 5 integers:");
for (int i = 0; i < arr.length; ++i) {
arr[i] = sc.nextInt();
}
}
// Method to print the state of the array
public static void printRecord(int[] arr) {
System.out.println("Array contents:");
for (int i = 0; i < arr.length; i++) {
System.out.println(arr[i]);
}
}
// Method to remove duplicate elements from the array
public static int[] removeDuplicates(int[] arr) {
int n = arr.length;
int[] temp = new int[n];
int k = 0;
// Check for duplicates and store unique elements in temp
for (int i = 0; i < n; i++) {
boolean isDuplicate = false;
for (int j = 0; j < k; j++) {
if (arr[i] == temp[j]) {
isDuplicate = true;
break;
}
}
if (!isDuplicate) {
temp[k++] = arr[i];
}
}
// Copy unique elements to a new array of size k
int[] uniqueArr = new int[k];
for (int i = 0; i < k; i++) {
uniqueArr[i] = temp[i];
}
return uniqueArr;
}
}
public class Q4 {
public static void main(String[] args) {
```

```
int[] arr = new int[5];
// Accept input into the array using RemoveDuplicates class
RemoveDuplicates.acceptRecord(arr);
// Print the original array using RemoveDuplicates class
System.out.println("Original array:");
RemoveDuplicates.printRecord(arr);
// Remove duplicates using RemoveDuplicates class
int[] uniqueArr = RemoveDuplicates.removeDuplicates(arr);
// Print the array after removing duplicates using RemoveDuplicates class
System.out.println("Array after removing duplicates:");
RemoveDuplicates.printRecord(uniqueArr);
}
}
```

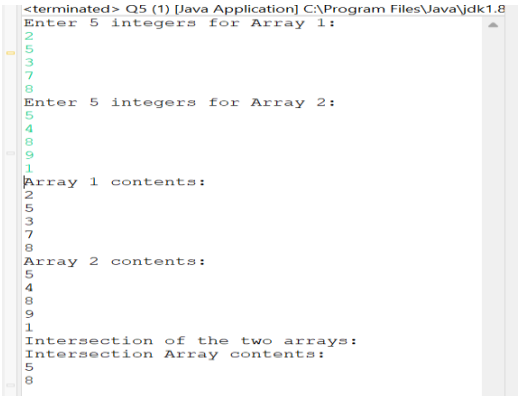


5. Write a program to find the intersection of two single-dimensional arrays.

```
package Assignment_6;

import java.util.Scanner;

import java.util.HashSet;

import java.util.Set;

class ArrayIntersection {

// Method to get input from the terminal into an array

public static void acceptRecord(int[] arr, String arrayName) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter 5 integers for " + arrayName + ":");

for (int i = 0; i < arr.length; ++i) {

arr[i] = sc.nextInt();
```

```
}

}

// Method to print the state of an array

public static void printRecord(int[] arr, String arrayName) {

System.out.println(arrayName + " contents:");

for (int i = 0; i < arr.length; i++) {

System.out.println(arr[i]);

}

}

// Method to find the intersection of two arrays

public static int[] findIntersection(int[] arr1, int[] arr2) {

Set<Integer> set1 = new HashSet<>();

Set<Integer> intersectionSet = new HashSet<>();

// Add elements of the first array to set1

for (int num : arr1) {

set1.add(num);

}

// Find intersection with the second array

for (int num : arr2) {

if (set1.contains(num)) {

intersectionSet.add(num);

}

}

// Convert the set to an array

int[] intersectionArr = new int[intersectionSet.size()];
```

```
int index = 0;

for (int num : intersectionSet) {

intersectionArr[index++] = num;

}

return intersectionArr;

}

}

public class Q5 {

public static void main(String[] args) {

int[] arr1 = new int[5];

int[] arr2 = new int[5];

// Accept input into the arrays

ArrayIntersection.acceptRecord(arr1, "Array 1");

ArrayIntersection.acceptRecord(arr2, "Array 2");

// Print the original arrays

ArrayIntersection.printRecord(arr1, "Array 1");

ArrayIntersection. printRecord(arr2, "Array 2");

// Find and print the intersection of the two arrays

int[] intersectionArr = ArrayIntersection.findIntersection(arr1, arr2);

System.out.println("Intersection of the two arrays:");

ArrayIntersection. printRecord(intersectionArr, "Intersection Array");

}

}
```



6. Write a program to find the missing number in an array of integers ranging from 1 to N.

```
package Assignment_6;
import java.util.Scanner;
//missing numbers
public class Q6 {
public static void acceptRecord(int[] arr) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter " + arr.length + " integers:");
for (int i = 0; i < arr.length; ++i) {
arr[i] = sc.nextInt();
}
}
// Method to find the missing number in an array of integers from 1 to N
public static int findMissingNumber(int[] arr, int N) {
// Calculate the sum of the first N natural numbers
int expectedSum = N * (N + 1) / 2;
// Calculate the sum of the elements in the array
int actualSum = 0;
for (int num : arr) {
actualSum += num;
}
// The missing number is the difference between expectedSum and actualSum
return expectedSum - actualSum;
}
public static void main(String[] args) {
int N = 5; // Example N value; you can change this based on your needs
int[] arr = new int[N - 1]; // Array size is N-1 because one number is missing
// Accept input into the array
acceptRecord(arr);
// Find and print the missing number
int missingNumber = findMissingNumber(arr, N);
System.out.println("The missing number is: " + missingNumber);
}
}
```

```
<terminated> Q6 [Java Application] C:\Program Files\Ja
Enter 4 integers:
12
43
76
90
The missing number is: -206
```

7. Declare a single-dimensional array as a field inside a class and instantiate it inside the class constructor. Define methods named `acceptRecord` and `printRecord` within the class and test their functionality.

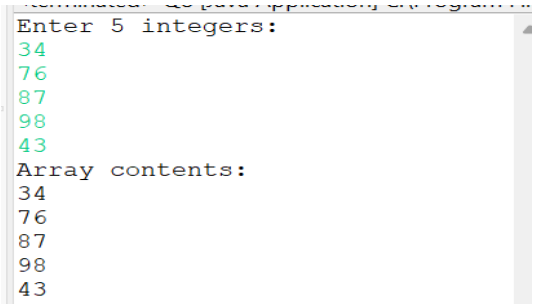
```
package Assignment_6;
import java.util.Scanner;
class ArrayHandler {
// Field to hold the array
private int[] arr;
// Constructor to initialize the array
public ArrayHandler(int size) {
arr = new int[size];
}
// Method to get input from the terminal into the array
public void acceptRecord() {
Scanner sc = new Scanner(System.in);
System.out.println("Enter " + arr.length + " integers:");
for (int i = 0; i < arr.length; ++i) {
arr[i] = sc.nextInt();
}
}
// Method to print the state of the array
public void printRecord() {
System.out.println("Array contents:");
for (int i = 0; i < arr.length; i++) {
System.out.println(arr[i]);
}
}
}
public class Q7 {
public static void main(String[] args) {
ArrayHandler arrayHandler = new ArrayHandler(5);
// Accept input into the array
arrayHandler.acceptRecord();
// Print the contents of the array
arrayHandler.printRecord();
}
}
```

```
<terminated> Q7 [Java Application] C:\Program Files\Java\jdk1.
Enter 5 integers:
45
87
90
85
34
Array contents:
45
87
90
85
34
```

8. Modify the previous assignment to use getter and setter methods instead of `acceptRecord` and `printRecord`.

```
package Assignment_6;
import java.util.Scanner;
class ArrHandler {
// Field to hold the array
private int[] arr;
// Constructor to initialize the array
public ArrHandler(int size) {
arr = new int[size];
}
// Setter method to populate the array
public void setArray(int[] arr) {
if (arr.length == this.arr.length) {
this.arr = arr;
} else {
System.out.println("Array size mismatch.");
}
}
// Getter method to retrieve the array
public int[] getArray() {
return arr;
}
// Method to print the state of the array
public void printRecord() {
System.out.println("Array contents:");
for (int i = 0; i < arr.length; i++) {
System.out.println(arr[i]);
}
}
}
public class Q8 {
public static void main(String[] args) {
// TODO Auto-generated method stub
ArrHandler arrayHandler = new ArrHandler(5);
int[] userArray = new int[5];
// Accept input from the user
Scanner sc = new Scanner(System.in);
System.out.println("Enter 5 integers:");
for (int i = 0; i < userArray.length; ++i) {
```

```
userArray[i] = sc.nextInt();
}
// Use setter method to set the array
arrayHandler.setArray(userArray);
// Print the contents of the array using the printRecord method
arrayHandler.printRecord();
}
}
```



9. You need to implement a system to manage airplane seat assignments. The airplane has seats arranged in rows and columns. Implement functionalities to:

   Initialize the seating arrangement with a given number of rows and columns.

   Book a seat to mark it as occupied.

   Cancel a booking to mark a seat as available.

   Check seat availability to determine if a specific seat is available.

   Display the current seating chart.

```
package Assignment_6;
import java.util.Scanner;
public class Q9 {
static class Airplane {
private boolean[][] seats; // 2D array to represent the seating arrangement
// Constructor to initialize the seating arrangement
public Airplane(int rows, int cols) {
seats = new boolean[rows][cols]; // false represents available seats
}
// Method to book a seat
public void bookSeat(int row, int col) {
if (isValidSeat(row, col)) {
if (seats[row][col]) {
System.out.println("Seat (" + row + ", " + col + ") is already booked.");
} else {
seats[row][col] = true;
System.out.println("Seat (" + row + ", " + col + ") booked successfully.");
```

```
}
} else {
System.out.println("Invalid seat number.");
}
}
// Method to cancel a booking
public void cancelBooking(int row, int col) {
if (isValidSeat(row, col)) {
if (!seats[row][col]) {
System.out.println("Seat (" + row + ", " + col + ") is not booked.");
} else {
seats[row][col] = false;
System.out.println("Booking for seat (" + row + ", " + col + ") canceled.");
}
} else {
System.out.println("Invalid seat number.");
}
}
// Method to check seat availability
public boolean isSeatAvailable(int row, int col) {
if (isValidSeat(row, col)) {
return !seats[row][col];
} else {
System.out.println("Invalid seat number.");
return false;
}
}
// Method to display the seating chart
public void displaySeatingChart() {
System.out.println("Seating Chart:");
for (int row = 0; row < seats.length; row++) {
for (int col = 0; col < seats[row].length; col++) {
if (seats[row][col]) {
System.out.print("[X] ");
} else {
System.out.print("[O] ");
}
}
System.out.println();
}
}
// Helper method to validate seat number
private boolean isValidSeat(int row, int col) {
return row >= 0 && row < seats.length && col >= 0 && col < seats[row].length;
}
}
public static void main(String[] args) {
```

```
Airplane airplane = new Airplane(5, 4);
// Create a scanner for user input
Scanner sc = new Scanner(System.in);
while (true) {
System.out.println("\n1. Book a seat");
System.out.println("2. Cancel a booking");
System.out.println("3. Check seat availability");
System.out.println("4. Display seating chart");
System.out.println("5. Exit");
System.out.print("Choose an option: ");
int option = sc.nextInt();
if (option == 1) {
System.out.print("Enter row (0-based index): ");
int row = sc.nextInt();
System.out.print("Enter column (0-based index): ");
int col = sc.nextInt();
airplane.bookSeat(row, col);
} else if (option == 2) {
System.out.print("Enter row (0-based index): ");
int row = sc.nextInt();
System.out.print("Enter column (0-based index): ");
int col = sc.nextInt();
airplane.cancelBooking(row, col);
} else if (option == 3) {
System.out.print("Enter row (0-based index): ");
int row = sc.nextInt();
System.out.print("Enter column (0-based index): ");
int col = sc.nextInt();
boolean available = airplane.isSeatAvailable(row, col);
System.out.println("Seat (" + row + ", " + col + ") is " + (available ? "available" : "not
available") + ".");
} else if (option == 4) {
airplane.displaySeatingChart();
} else if (option == 5) {
break;
} else {
System.out.println("Invalid option. Please try again.");
}
}
sc.close();
}
}
```