

Arrays and Functions

1. Take 20 integer inputs from user and print the following:

1. number of positive numbers
2. number of negative numbers
3. number of odd numbers
4. number of even numbers
5. number of 0s.

Sample Input:

```
1 -3 6 9 8 -13 -5 7 0 12 0 -4 4 0 17 21 6 16 11 19
```

Sample Output:

```
13 4 10 10 3
```

Analysis:

1. Get the number of integers from the user.
2. Initialize the variables for positive, negative, odd, even, and zero numbers.
3. Loop through the numbers and check if the number is positive, negative, odd, even, or zero.
4. Increment the respective counter.
5. Print the number of positive, negative, odd, even, and zero numbers.

Program: NumberCounter.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class DefineNumber {
```

```
public void countPositive(int[] arr){
    int positive = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i] > 0){
            positive++;
        }
    }
    System.out.println("Count of positive number: "+positive);
}
```

```
public void countNegative(int[] arr){
    int negative = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i] < 0){
            negative++;
        }
    }
    System.out.println("Count of negative number: "+negative);
}
```

```
public void countOdd(int[] arr){
    int odd = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i] % 2 != 0){
            odd++;
        }
    }
    System.out.println("Count of odd number: "+odd);
}
```

```
public void countEven(int[] arr){
    int even = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i] % 2 == 0){
            even++;
        }
    }
    System.out.println("Count of even number: "+even);
}
```

```
public void countZero(int[] arr){
    int zero = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i] == 0){
```

```

        zero++;
    }
}

System.out.println("Count of zero: "+zero);
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the Array size: ");
    int arraySize = sc.nextInt();
    System.out.print("Enter the Array Elements: ");
    int[] arr = new int[arraySize];
    for(int i=0; i<arraySize; i++){
        arr[i] = sc.nextInt();
    }

    DefineNumber dn = new DefineNumber();
    dn.countPositive(arr);
    dn.countNegative(arr);
    dn.countOdd(arr);
    dn.countEven(arr);
    dn.countZero(arr);

    sc.close();
}
}

```

Output:

```

Enter the Array size: 10
Enter the Array Elements: 1 -3 6 9 8 -13 -5 7 0 12
Count of positive number: 6
Count of negative number: 3
Count of odd number: 5
Count of even number: 5
Count of zero: 1

```

2. There is long queue in the billing counter of a supermarket. Tell the position of the specific customer if the names are the input. If not found, print -1.

Sample Input:

```
5
Smith Tim Eve John Dora
Eve
```

Sample Output:

```
3
```

Sample Input:

```
5
Smith Tim Eve John Dora
Mike
```

Sample Output:

```
-1
```

Analysis:

1. Get the number of customers from the user.
2. Get the names of the customers.
3. Get the name of the customer to find the position.
4. Loop through the names and check if the name matches the name to find the position.
5. Print the position if found, else print -1.

Program: `CustomerPosition.java`

Code:

```

package com.hands_on;

import java.util.Scanner;

public class CustomerPosition {
    // using function to find the position of the customer
    public int findPosition(String[] names, String customer){
        for(int i=0; i<names.length; i++){
            if(names[i].equals(customer)){
                return i+1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of customers: ");
        int n = sc.nextInt();
        sc.nextLine();
        String[] names = new String[n];
        System.out.println("Enter the customer names: ");
        for(int i=0; i<n; i++){
            names[i] = sc.nextLine();
        }

        System.out.print("Enter the customer name to find the position: ");
        String customer = sc.nextLine();

        CustomerPosition cp = new CustomerPosition();
        int position = cp.findPosition(names, customer);
        System.out.println("The Position of "+customer+ " is "+ position);

        sc.close();
    }
}

```

Output:

```
Enter the number of customers: 5
Enter the customer names:
subasri
shabari
sowandharya
sathyaram
manoj
Enter the customer name to find the position: shabari
The Position of shabari is 2
```

3. Given an array of size N-1 such that it only contains distinct integers in the range of 1 to N. Find the missing element. Assume that integer range is correctly given.

Sample Input:

```
5, [1,2,3,5]
```

Sample Output:

```
4
```

Sample Input:

```
10, [6,1,2,8,3,4,7,10,5]
```

Sample Output:

```
9
```

Analysis:

1. Get the integer range and the array of distinct integers.
2. Find the sum of the integers in the range.

3. Find the sum of the integers in the array.
4. Subtract the sum of the integers in the array from the sum of the integers in the range.
5. Print the missing element.

Program: MissingElement.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class MissingElement {
    public int findMissingElement(int n, int[] arr){
        int sum = n*(n+1)/2;
        int sumArr = 0;
        for(int i=0; i<arr.length; i++){
            sumArr += arr[i];
        }
        return sum - sumArr;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the integer range: ");
        int n = sc.nextInt();
        int[] arr = new int[n-1];
        System.out.println("Enter the array elements: ");
        for(int i=0; i<n-1; i++){
            arr[i] = sc.nextInt();
        }

        MissingElement me = new MissingElement();
        int missingElement = me.findMissingElement(n, arr);
        System.out.println("The Missing Element is: "+missingElement);

        sc.close();
    }
}
```

Output:

```
Enter the integer range: 10
Enter the array elements:
6 1 2 8 3 4 7 10 5
The Missing Element is: 9
```

4. Write a program to find & remove duplicate elements in the array and reprint.

Sample Input:

```
1 2 8 3 4 5 5 6 7 8
```

Sample Output:

```
1 2 8 3 4 5 6 7
```

Analysis:

1. Get the array of integers from the user.
2. Initialize a new array to store the unique elements.
3. Loop through the array and check if the element is already in the new array.
4. If not, add the element to the new array.
5. Print the new array.

Program: RemoveDuplicates.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class RemoveDuplicates {
```



```

public void removeDuplicates(int[] arr){
    int[] uniqueArr = new int[arr.length];
    int k = 0;
    for(int i=0; i<arr.length; i++){
        int j;
        for(j=0; j<i; j++){
            if(arr[i] == arr[j]){
                break;
            }
        }
        if(i == j){
            uniqueArr[k++] = arr[i];
        }
    }
    System.out.println("Array after removing duplicates: ");
    for(int i=0; i<k; i++){
        System.out.print(uniqueArr[i]+" ");
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the Array size: ");
    int arraySize = sc.nextInt();
    int[] arr = new int[arraySize];
    System.out.println("Enter the Array Elements: ");
    for(int i=0; i<arraySize; i++){
        arr[i] = sc.nextInt();
    }

    RemoveDuplicates rd = new RemoveDuplicates();
    rd.removeDuplicates(arr);

    sc.close();
}
}

```

Output:

```

Enter the Array size: 9
Enter the Array Elements:

```

```
1 2 8 3 4 5 5 6 7 8
```

Array after removing duplicates:

```
1 2 8 3 4 5 6 7
```

5. Mike enters randomly twenty-five numbers from the keyboard and stores it into an array. He wants to search if the number is present in the array and if it is present, he needs to display the number of times it appears in the array. Help him with your program.

Sample Input:

```
1 2 8 3 4 5 5 6 7 8 1 2 8 3 4 5 5 6 7 8 1 2 8 3 4
8
```

Sample Output:

```
5
```

Analysis:

1. Get the array of integers from the user.
2. Get the number to search from the user.
3. Initialize a counter to count the number of times the number appears in the array.
4. Loop through the array and check if the number matches the number to search.
5. If found, increment the counter.
6. Print the counter.

Program: NumberFrequency.java

Code:

```
package com.hands_on;
```

```

import java.util.Scanner;

public class NumberFrequency {
    public void findFrequency(int[] arr, int num){
        int count = 0;
        for(int i=0; i<arr.length; i++){
            if(arr[i] == num){
                count++;
            }
        }
        System.out.println("The Frequency of "+num+" is: "+count);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Array size: ");
        int arraySize = sc.nextInt();
        int[] arr = new int[arraySize];
        System.out.println("Enter the Array Elements: ");
        for(int i=0; i<arraySize; i++){
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter the number to search: ");
        int num = sc.nextInt();

        NumberFrequency nf = new NumberFrequency();
        nf.findFrequency(arr, num);

        sc.close();
    }
}

```

Output:

```

Enter the Array size: 10
Enter the Array Elements:
1 2 3 3 3 4 1 2 3 3
Enter the number to search: 3
The Frequency of 3 is: 5

```

6. Write a program to find pair of elements in the array having sum of 10. If not found any, return -1.

Sample Input:

```
1 2 8 3
```

Sample Output:

```
(2,8)
```

Sample Input:

```
1 2 3 4 5
```

Sample Output:

```
-1
```

Analysis:

1. Get the array of integers from the user.
2. Initialize a variable to check if the pair is found.
3. Loop through the array and check if the sum of the pair is 10.
4. If found, print the pair and set the variable to true.
5. If not found, print -1.

Program: PairSum.java

Code:

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

```

public class PairSum {
    public void findPair(int[] arr){
        boolean found = false;
        for(int i=0; i<arr.length; i++){
            for(int j=i+1; j<arr.length; j++){
                if(arr[i] + arr[j] == 10){
                    System.out.println("(" + arr[i] + ", " + arr[j] + ")");
                    found = true;
                }
            }
        }
        if(!found){
            System.out.println("-1");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Array size: ");
        int arraySize = sc.nextInt();
        int[] arr = new int[arraySize];
        System.out.println("Enter the Array Elements: ");
        for(int i=0; i<arraySize; i++){
            arr[i] = sc.nextInt();
        }

        PairSum ps = new PairSum();
        ps.findPair(arr);

        sc.close();
    }
}

```

Output:

```

Enter the Array size: 4
Enter the Array Elements:
1 2 8 3
(2,8)

```

7. Write a program to accept and print elements in 2D array. Get the row and column size from the user.

Sample Input:

```
3,4, [1 2 3 4 5 6 7 8 9 10 11 12]
```

Sample Output:

```
1  2  3  4
5  6  7  8
9 10 11 12
```

Analysis:

1. Get the row and column size from the user.
2. Initialize a 2D array with the row and column size.
3. Get the elements of the 2D array from the user.
4. Print the 2D array.

Program: TwoDArray.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class TwoDArray {
    void display(int[][] arr){
        System.out.println("The Two-Dimensional Array: ");
        for(int row=0; row<arr.length; row++){
            for(int col=0; col<arr[row].length; col++){
                System.out.print(arr[row][col]+" ");
            }
            System.out.println();
        }
    }
}
```

```

    }
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the row size: ");
    int rowSize = sc.nextInt();
    System.out.print("Enter the column size: ");
    int columnSize = sc.nextInt();
    System.out.println("Enter the array elements: ");
    int[][] array = new int[rowSize][columnSize];
    for(int row=0; row<rowSize; row++){
        for(int col=0; col<columnSize; col++){
            array[row][col] = sc.nextInt();
        }
    }
    TwoDArray twodarr = new TwoDArray();
    twodarr.display(array);
}
}

```

Output:

```

Enter the row size: 3
Enter the column size: 3
Enter the array elements:
1 2 3
4 5 6
7 8 9
The Two-Dimensional Array:
1 2 3
4 5 6
7 8 9

```

8. Write a Java program to replace each element of the array with product of all other elements in a given array of integers.

Sample Input:

```
4, [1 2 3 4]
```

Sample Output:

```
24 12 8 6
```

Analysis:

1. Get the array of integers from the user.
2. Initialize a new array to store the product of all other elements.
3. Loop through the array and find the product of all other elements.
4. Print the new array.

Program: ProductOfElements.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class ProductOfElements {
    public void findProduct(int[] arr){
        int[] productArr = new int[arr.length];
        for(int i=0; i<arr.length; i++){
            int product = 1;
            for(int j=0; j<arr.length; j++){
                if(i != j){
                    product *= arr[j];
                }
            }
            productArr[i] = product;
        }
        System.out.println("The Product of all other elements: ");
        for(int i=0; i<productArr.length; i++){
            System.out.print(productArr[i]+" ");
        }
    }

    public static void main(String[] args) {
```



```

Scanner sc = new Scanner(System.in);

System.out.print("Enter the Array size: ");
int arraySize = sc.nextInt();
int[] arr = new int[arraySize];
System.out.println("Enter the Array Elements: ");
for(int i=0; i<arraySize; i++){
    arr[i] = sc.nextInt();
}

ProductOfElements poe = new ProductOfElements();
poe.findProduct(arr);

sc.close();
}
}

```

Output:

```

Enter the Array size: 4
Enter the Array Elements:
1 2 3 4
The Product of all other elements:
24 12 8 6

```

9. Get the values for an array of size 10. Write the logic to find whether the array elements are in **Arithmetic Progression** or **Geometric Progression**. If the array is in neither order display **Random order**.

Sample Input:

```
1 4 7 10 13 16 19 22 25 28
```

Sample Output:

Arithmetic Progression

Sample Input:

1 2 4 8 16 32 64 128 256 512

Sample Output:

Geometric Progression

Sample Input:

2 4 7 11 16 22 29 37 46 56

Sample Output:

Random Order

Analysis:

1. Get the array of integers from the user.
2. Check if the array is in Arithmetic Progression.
3. Check if the array is in Geometric Progression.
4. If not, print Random Order.

Program: Progression.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class Progression {
    public void findProgression(int[] arr){
        boolean isArithmetic = true;
        boolean isGeometric = true;
```

```

        for(int i=2; i<arr.length; i++){
            if(arr[i] - arr[i-1] != arr[i-1] - arr[i-2]){
                isArithmetic = false;
            }
            if(arr[i] / arr[i-1] != arr[i-1] / arr[i-2]){
                isGeometric = false;
            }
        }
        if(isArithmetic){
            System.out.println("Arithmetic Progression");
        }else if(isGeometric){
            System.out.println("Geometric Progression");
        }else{
            System.out.println("Random Order");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Array size: ");
        int arraySize = sc.nextInt();
        int[] arr = new int[arraySize];
        System.out.println("Enter the Array Elements: ");
        for(int i=0; i<arraySize; i++){
            arr[i] = sc.nextInt();
        }

        Progression pr = new Progression();
        pr.findProgression(arr);

        sc.close();
    }
}

```

Output:

```

Enter the Array size: 10
Enter the Array Elements:
1 2 4 8 16 32 64 128 256 512
Geometric Progression

```

10. In a lucky draw, XYZ finance company selects two sets of its customers for a promotion. If the customer's coupon is in first set, then the customer gets Rs.10000/- as cash prize. If it is in second set, then the customer gets tour tickets for two days. Otherwise, customer gets a batch **Better luck next time** . Two sets of coupon numbers and a randomly picked customer coupon are the inputs. Help the company to say the result. Note: Consider each set has 10 distinctive customer coupons and no common coupons.

Sample Input:

```
[2 4 7 11 16 22 29 37 46 56]
[1 5 9 10 13 18 19 22 25 28]
16
```

Sample Output:

```
Rs.10000 Cash Prize
```

Sample Input:

```
[2 4 7 11 16 22 29 37 46 56]
[1 5 9 10 13 18 19 22 25 28]
13
```

Sample Output:

```
Tour Tickets for two days
```

Analysis:

1. Get the two sets of coupon numbers and the customer coupon number.
2. Check if the customer coupon number is in the first set.

3. If found, print Rs.10000 Cash Prize.
4. If not, check if the customer coupon number is in the second set.
5. If found, print Tour Tickets for two days.
6. If not found in both sets, print Better luck next time.

Program: LuckyDraw.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class LuckyDraw {
    public void findPrize(int[] set1, int[] set2, int customer){
        boolean found = false;
        for(int i=0; i<set1.length; i++){
            if(set1[i] == customer){
                System.out.println("Rs.10000 Cash Prize");
                found = true;
            }
        }
        if(!found){
            for(int i=0; i<set2.length; i++){
                if(set2[i] == customer){
                    System.out.println("Tour Tickets for two days");
                    found = true;
                }
            }
        }
        if(!found){
            System.out.println("Better luck next time");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the first set of coupon numbers: ");
        int[] set1 = new int[10];
        for(int i=0; i<10; i++){
            set1[i] = sc.nextInt();
        }
    }
}
```

```

    }

    System.out.println("Enter the second set of coupon numbers: ");
    int[] set2 = new int[10];
    for(int i=0; i<10; i++){
        set2[i] = sc.nextInt();
    }

    System.out.print("Enter the customer coupon number: ");
    int customer = sc.nextInt();

    LuckyDraw ld = new LuckyDraw();
    ld.findPrize(set1, set2, customer);

    sc.close();
}
}

```

Output:

```

Enter the first set of coupon numbers:
2 4 7 11 16 22 29 37 46 56
Enter the second set of coupon numbers:
1 5 9 10 13 18 19 22 25 28
Enter the customer coupon number: 16
Rs.10000 Cash Prize

```

11. XYZ College asked their students to register for NSS and NCC if they are willing. Some of the students registered for both. Identify them if student ids(numeric) for each group is the input.

Sample Input:

```

10
10

```

```
2 4 7 11 16 22 29 37 46 56
1 4 7 10 13 16 19 22 25 28
```

Sample Output:

```
4 7 16 22
```

Analysis:

1. Get the number of students and the student ids for NSS and NCC.
2. Initialize a new array to store the student ids who registered for both.
3. Loop through the student ids for NSS and check if the student id is in the student ids for NCC.
4. If found, add the student id to the new array.
5. Print the student ids who registered for both.

Program: RegisterStudents.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class RegisterStudents {
    public void findCommonStudents(int n, int[] nss, int[] ncc){
        System.out.println("Students who registered for both NSS and NCC: ");
        for(int i=0; i<n; i++){
            for(int j=0; j<n; j++){
                if(nss[i] == ncc[j]){
                    System.out.print(nss[i]+" ");
                }
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
```

```

        System.out.println("Enter the student ids for NSS: ");
        int[] nss = new int[n];
        for(int i=0; i<n; i++){
            nss[i] = sc.nextInt();
        }

        System.out.println("Enter the student ids for NCC: ");
        int[] ncc = new int[n];
        for(int i=0; i<n; i++){
            ncc[i] = sc.nextInt();
        }

        RegisterStudents rs = new RegisterStudents();
        rs.findCommonStudents(n, nss, ncc);

        sc.close();
    }
}

```

Output:

```

Enter the number of students: 10
Enter the student ids for NSS:
2 4 7 11 16 22 29 37 46 56
Enter the student ids for NCC:
1 4 7 10 13 16 19 22 25 28
Students who registered for both NSS and NCC:
4 7 16 22

```

12. Split an array in to two arrays such that one array contains the elements lesser than the average of the given array and the other contains the greater numbers. Skip the element if it is equal to the average.

Sample Input:

5

[10 20 50 30 45]

Sample Output:

[10 20 30]

[50 45]

Analysis:

1. Get the number of elements and the array of integers from the user.
2. Find the average of the array.
3. Initialize two new arrays to store the elements lesser and greater than the average.
4. Loop through the array and check if the element is lesser or greater than the average.
5. If equal, skip the element.
6. Print the two arrays.

Program: SplitArray.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class SplitArray {
    public void splitArray(int n, int[] arr){
        int sum = 0;
        for(int i=0; i<n; i++){
            sum += arr[i];
        }
        int avg = sum/n;
        int[] lesserArr = new int[n];
        int[] greaterArr = new int[n];
        int lesserIndex = 0;
        int greaterIndex = 0;
        for(int i=0; i<n; i++){
            if(arr[i] < avg){
                lesserArr[lesserIndex++] = arr[i];
            }
        }
    }
}
```



```

        }else if(arr[i] > avg){
            greaterArr[greaterIndex++] = arr[i];
        }
    }
    System.out.println("Array of elements lesser than average: ");
    for(int i=0; i<lesserIndex; i++){
        System.out.print(lesserArr[i]+" ");
    }
    System.out.println();
    System.out.println("Array of elements greater than average: ");
    for(int i=0; i<greaterIndex; i++){
        System.out.print(greaterArr[i]+" ");
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of elements: ");
    int n = sc.nextInt();

    System.out.println("Enter the array elements: ");
    int[] arr = new int[n];
    for(int i=0; i<n; i++){
        arr[i] = sc.nextInt();
    }

    SplitArray sa = new SplitArray();
    sa.splitArray(n, arr);

    sc.close();
}
}

```

Output:

```

Enter the number of elements: 5
Enter the array elements:
10 20 50 30 45
Array of elements lesser than average:
10 20 30

```

Array of elements greater than average:

50 45

13. Heena and Reena bet for a chocolate with the number game. If Heena fails to say a group of numbers in the order which is same even it is read backward, then Reena wins. Write a program to check if the given array of numbers is palindromic or not.

Sample Input:

```
5
1 2 3 2 1
```

Sample Output:

```
true
```

Analysis:

1. Get the number of elements and the array of integers from the user.
2. Initialize two pointers to the start and end of the array.
3. Check if the element at the start and end are equal.
4. If equal, increment the start pointer and decrement the end pointer.
5. If not equal, print false.
6. If the loop completes, print true.

Program: PalindromicArray.java

Code:

```
package com.hands_on;

import java.util.Scanner;
```

```

public class PalindromicArray {
    public boolean isPalindromic(int n, int[] arr){
        int start = 0;
        int end = n-1;
        while(start < end){
            if(arr[start] != arr[end]){
                return false;
            }
            start++;
            end--;
        }
        return true;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();

        System.out.println("Enter the array elements: ");
        int[] arr = new int[n];
        for(int i=0; i<n; i++){
            arr[i] = sc.nextInt();
        }

        PalindromicArray pa = new PalindromicArray();
        boolean isPalindromic = pa.isPalindromic(n, arr);
        System.out.println(isPalindromic);

        sc.close();
    }
}

```

Output:

```

Enter the number of elements: 5
Enter the array elements:
1 2 3 2 1
true

```

14. Write a Java program to accept N numbers from console and print the sum of the elements of the array with the following condition. Condition: If the array has elements **a** and **b** in succeeding order, ignore the numbers between **a** and **b** inclusive for the calculation.

Sample Input:

```
7
10,3,6,1,2,7,9
6
7
```

Sample Output:

```
22
```

Sample Input:

```
5
7,1,2,3,6
6
7
```

Sample Output:

```
19
```

Sample Input:

```
4
1,6,7,9
6
7
```

Sample Output:

10

Analysis:

1. Get the number of elements and the array of integers from the user.
2. Get the elements `a` and `b` from the user.
3. Initialize a variable to store the sum of the elements.
4. Loop through the array and check if the element is `a`.
5. If found, skip the elements until `b`.
6. Print the sum of the elements.

Program: SumOfElements.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class SumOfElements {
    public int findSum(int n, int[] arr, int a, int b){
        int sum = 0;
        boolean skip = false;
        for(int i=0; i<n; i++){
            if(arr[i] == a){
                skip = true;
            }else if(arr[i] == b){
                skip = false;
            }else if(!skip){
                sum += arr[i];
            }
        }
        return sum;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```

        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();

        System.out.println("Enter the array elements: ");
        int[] arr = new int[n];
        for(int i=0; i<n; i++){
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter the element a: ");
        int a = sc.nextInt();
        System.out.print("Enter the element b: ");
        int b = sc.nextInt();

        SumOfElements soe = new SumOfElements();
        int sum = soe.findSum(n, arr, a, b);
        System.out.println(sum);

        sc.close();
    }
}

```

Output:

```

Enter the number of elements: 7
Enter the array elements:
10 3 6 1 2 7 9
Enter the element a: 6
Enter the element b: 7
22

```

15. There are N friends in a group. Each of them has X_i chocolates. Write a Java Program to check whether they can share all these chocolates among themselves such that each one of them has equal number of chocolates. Input: First line contains of a single line of input, an integer N denoting no. of

friends in the group. Next line contains N space separated integers X_i denoting the no. chocolates ith friend has. Output "Yes" if it is possible to share equally else "No" (without " ").

Sample Input:

```
3
1 2 3
```

Sample Output:

```
Yes
```

Analysis:

1. Get the number of friends and the number of chocolates each friend has from the user.
2. Find the total number of chocolates.
3. Check if the total number of chocolates is divisible by the number of friends.
4. If divisible, print Yes. Else, print No.

Program: `ShareChocolates.java`

Code:

```
package com.hands_on;

import java.util.Scanner;

public class ShareChocolates {
    public String canShare(int n, int[] arr){
        int totalChocolates = 0;
        for(int i=0; i<n; i++){
            totalChocolates += arr[i];
        }
        if(totalChocolates % n == 0){
            return "Yes";
        }
        return "No";
    }
}
```



```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of friends: ");
    int n = sc.nextInt();

    System.out.println("Enter the number of chocolates each friend has: ");
    int[] arr = new int[n];
    for(int i=0; i<n; i++){
        arr[i] = sc.nextInt();
    }

    ShareChocolates sch = new ShareChocolates();
    String result = sch.canShare(n, arr);
    System.out.println(result);

    sc.close();
}
}

```

Output:

```

Enter the number of friends: 3
Enter the number of chocolates each friend has:
1 2 3
Yes

```

16. Write a program to find the minimum and maximum element of each row and column in the given two-dimensional arrays.

Sample Input:

```

Enter row size: 3
Enter column size: 3
Enter 3 * 3 array elements are: 4

```

```
1
2
5
3
6
3
7
8
```

Sample Output:

Given Array is:

```
4 1 2
5 3 6
3 7 8
```

The minimum and maximum element of 1st row is: 1, 4

The minimum and maximum element of 2nd row is: 3, 6

The minimum and maximum element of 3rd row is: 3, 8

The minimum and maximum element of 1st column is: 3,
5

The minimum and maximum element of 2nd column is: 1,
7

The minimum and maximum element of 3rd column is: 2,
8

Analysis

1. Get the row and column size from the user.
2. Get the elements of the 2D array from the user.
3. Print the 2D array.
4. Find the minimum and maximum element of each row and column.
5. Print the minimum and maximum element of each row and column.

Code:

```
package com.hands_on;

import java.util.Scanner;

public class MinMaxElements {
```

```

public void findMinMax(int row, int col, int[][] arr){
    System.out.println("Given Array is: ");
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            System.out.print(arr[i][j]+" ");
        }
        System.out.println();
    }
    for(int i=0; i<row; i++){
        int minRow = Integer.MAX_VALUE;
        int maxRow = Integer.MIN_VALUE;
        for(int j=0; j<col; j++){
            if(arr[i][j] < minRow){
                minRow = arr[i][j];
            }
            if(arr[i][j] > maxRow){
                maxRow = arr[i][j];
            }
        }
        System.out.println(
            "The minimum and maximum element of "
            + ( i+1 ) + " row is: "
            + minRow + ", " + maxRow
        );
    }
    for(int i=0; i<col; i++){
        int minCol = Integer.MAX_VALUE;
        int maxCol = Integer.MIN_VALUE;
        for(int j=0; j<row; j++){
            if(arr[j][i] < minCol){
                minCol = arr[j][i];
            }
            if(arr[j][i] > maxCol){
                maxCol = arr[j][i];
            }
        }
        System.out.println(
            "The minimum and maximum element of "
            + ( i+1 ) + " column is: "
            + minCol + ", " + maxCol
        );
    }
}

```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter row size: ");
    int row = sc.nextInt();
    System.out.print("Enter column size: ");
    int col = sc.nextInt();

    int[][] arr = new int[row][col];
    System.out.println(
        "Enter "
        + row + " * " + col
        + " array elements are: "
    );
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            arr[i][j] = sc.nextInt();
        }
    }

    MinMaxElements mme = new MinMaxElements();
    mme.findMinMax(row, col, arr);

    sc.close();
}
}

```

17. Write a Java program to accept n numbers from console. Store all input numbers in the array. When the negative number is entered, the negative number is ignored and input stops.

1. Replace the numbers in array as per following rules:
2. Replace a number in array with 0 if it is even.
3. Replace a number in array with 1 if it is odd.
4. Replace a number in array with 2 if it is divisible by 8.
5. Replace a number in array with 3 if it ends with 3.

6. Replace a number in array with 4 if it is divisible by 9.
7. If multiple rules apply to a number, use the rule that replaces with highest number
8. Print the array before and after replacing

Sample Input:

```
5 2 8 9 16 27 6 1 18 -12
```

Sample Output:

```
5 2 8 9 16 27 6 1 18
1 0 2 4 2 4 3 1 4
```

Analysis

1. Get the user input numbers and store them in an array.
2. Replace the numbers in the array based on the rules.
3. Print the array before and after replacing.

Program: ReplaceNumber.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class ReplaceNumber {
    public void replaceNumbers(int n, int[] arr){
        System.out.println("Array before replacing: ");
        for(int i=0; i<n; i++){
            System.out.print(arr[i]+" ");
        }
        System.out.println();
        for(int i=0; i<n; i++){
            if(arr[i] < 0){
                continue;
            }
            if(arr[i] % 9 == 0){
                arr[i] = 4;
            }
        }
    }
}
```

```

        }else if(arr[i] % 8 == 0){
            arr[i] = 2;
        }else if(arr[i] % 2 == 0){
            arr[i] = 0;
        }else if(arr[i] % 10 == 3){
            arr[i] = 3;
        }else{
            arr[i] = 1;
        }
    }
    System.out.println("Array after replacing: ");
    for(int i=0; i<n; i++){
        System.out.print(arr[i]+" ");
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter the numbers: ");
    int[] arr = new int[100];
    int n = 0;
    while(true){
        int num = sc.nextInt();
        if(num < 0){
            break;
        }
        arr[n++] = num;
    }

    ReplaceNumber rn = new ReplaceNumber();
    rn.replaceNumbers(n, arr);

    sc.close();
}
}

```

Output:

```

Enter the numbers:
5 2 8 9 16 27 6 1 18 -12
Array before replacing:

```

```
5 2 8 9 16 27 6 1 18
Array after replacing:
1 0 2 4 2 4 3 1 4
```

18. Consider an integer array, the number of elements in which is determined by the user. The elements are also taken as input from the user. Write a program to find those pair of elements that has the maximum and minimum difference among all element pairs.

Analysis

1. Get the number of elements and the array of integers from the user.
2. Initialize two variables to store the maximum and minimum difference.
3. Loop through the array and find the difference between each pair of elements.
4. If the difference is greater than the maximum difference, update the maximum difference.
5. If the difference is lesser than the minimum difference, update the minimum difference.
6. Print the pair of elements with the maximum and minimum difference.

Program: MaxMinDifference.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class MaxMinDifference {
    public void findDifference(int n, int[] arr){
        int maxDiff = Integer.MIN_VALUE;
        int minDiff = Integer.MAX_VALUE;
        int num1 = 0;
        int num2 = 0;
        for(int i=0; i<n; i++){
            for(int j=i+1; j<n; j++){
                int diff = Math.abs(arr[i] - arr[j]);
```

```

        if(diff > maxDiff){
            maxDiff = diff;
            num1 = arr[i];
            num2 = arr[j];
        }
        if(diff < minDiff){
            minDiff = diff;
        }
    }
}

System.out.println("The pair of elements with maximum difference: "+n
System.out.println("The minimum difference: "+minDiff);
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of elements: ");
    int n = sc.nextInt();

    System.out.println("Enter the array elements: ");
    int[] arr = new int[n];
    for(int i=0; i<n; i++){
        arr[i] = sc.nextInt();
    }

    MaxMinDifference mmd = new MaxMinDifference();
    mmd.findDifference(n, arr);

    sc.close();
}
}

```

Output:

```

Enter the number of elements: 5
Enter the array elements:
1 3 5 7 9
The pair of elements with maximum difference: 1, 9
The minimum difference: 2

```

19. Develop a Java program that constructs a jagged array for recording basketball tournament scores per player. The program initiates by requesting the user to input the count of teams along with the number of players within each team. Following this, it enables the user to input the scores for every player across each team. Lastly, the program computes and exhibits the average score for every team.

Analysis

1. Get the number of teams and the number of players in each team from the user.
2. Initialize a jagged array to store the scores of each player in each team.
3. Get the scores of each player from the user.
4. Calculate the average score of each team.

Program: `BasketballScores.java`

Code:

```
package com.hands_on;

import java.util.Scanner;

public class BasketballScores {
    public void findAverage(int n, int[][] arr){
        for(int i=0; i<n; i++){
            int sum = 0;
            int count = 0;
            for(int j=0; j<arr[i].length; j++){
                sum += arr[i][j];
                count++;
            }
            double avg = (double)sum/count;
            System.out.println("Average score of Team "+(i+1)+": "+avg);
        }
    }
}
```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of teams: ");
    int n = sc.nextInt();

    int[][] arr = new int[n][];
    for(int i=0; i<n; i++){
        System.out.print("Enter the number of players in Team "+(i+1)+":");
        int players = sc.nextInt();
        arr[i] = new int[players];
        System.out.println("Enter the scores of players in Team "+(i+1)+");
        for(int j=0; j<players; j++){
            arr[i][j] = sc.nextInt();
        }
    }

    BasketballScores bs = new BasketballScores();
    bs.findAverage(n, arr);

    sc.close();
}

```

Output:

```

Enter the number of teams: 3
Enter the number of players in Team 1: 3
Enter the scores of players in Team 1:
10
20
30
Enter the number of players in Team 2: 2
Enter the scores of players in Team 2:
15
25
Average score of Team 1: 20.0
Average score of Team 2: 20.0

```

20. You're developing a student grading system where each student has various assignments, each with its own weightage. Create a program that first asks the user to input the total number of students and the number of assignments. Then, enable the user to input grades for each assignment for every student. Finally, compute and present the weighted average grade for each student, factoring in the assignment weights.

Analysis

1. Get the number of students and the number of assignments from the user.
2. Initialize a 2D array to store the grades of each student for each assignment.
3. Get the grades of each student for each assignment from the user.
4. Calculate the weighted average grade for each student.

Program: StudentGrades.java

Code:

```
package com.hands_on;

import java.util.Scanner;

public class StudentGrades {
    public void findAverage(int n, int m, int[][] arr){
        System.out.println("Weighted Average Grades of Students: ");
        for(int i=0; i<n; i++){
            double sum = 0;
            for(int j=0; j<m; j++){
                sum += arr[i][j];
            }
            double avg = sum/m;
            System.out.println("Student "+(i+1)+": "+avg);
        }
    }

    public static void main(String[] args) {
```

```

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of students: ");
int n = sc.nextInt();

System.out.print("Enter the number of assignments: ");
int m = sc.nextInt();

int[][] arr = new int[n][m];
for(int i=0; i<n; i++){
    System.out.println("Enter the grades of Student "+(i+1)+" : ");
    for(int j=0; j<m; j++){
        arr[i][j] = sc.nextInt();
    }
}

StudentGrades sg = new StudentGrades();
sg.findAverage(n, m, arr);

sc.close();
}
}

```

Output:

```

Enter the number of students: 3
Enter the number of assignments: 2
Enter the grades of Student 1:
10
20
Enter the grades of Student 2:
15
25
Enter the grades of Student 3:
20
30
Weighted Average Grades of Students:
Student 1: 15.0
Student 2: 20.0
Student 3: 25.0

```

