

## Week 2 – Batch3

---

Q1: Given the float array, return the sum of all elements in it.

Program:

```
import java.util.Scanner;

public class Sum_Arr {

    Scanner sc = new Scanner(System.in);
    public int size;
    float arr[] = new float[100];
    public float s=0;
    void getele() {
        System.out.println("Enter the size of the array: ");
        size = sc.nextInt();
        System.out.println("Enter the elements: ");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextFloat();
        }
    }

    float getsum() {
        for (int i = 0; i < size; i++) {
            s = s + arr[i];
        }
        return s;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Sum_Arr obj = new Sum_Arr();
        obj.getele();
        float sum = obj.getsum();
        System.out.println("The sum of the elements of the array: " + sum);
    }
}
```

Output:

Array Size	Input	Expected Output	My Output
5	4,6,8,9,0	27.0	27.0
6	3,2,66,9,76,1,4	161.0	157.0
4	-6,6,-3,3	0	0
7	22,-6,43,-19,-40,-2,6	-2	4
6	a,b,c,1,4,7	Exception	Exception

---

Q2: Return the sum of the first, middle and last elements in the array if the length of the array is odd, if the length is even then the sum of the first and last element should be displayed. [Hint: import required java.util.Array package]

Program:

```
import java.util.Scanner;
import java.util.Arrays;
```

```

import java.util.InputMismatchException;
public class Arr_Sum_Len {

Scanner sc = new Scanner(System.in);
public int size, s;
int arr[]=new int[100];

void getdata(){
System.out.println("Enter the size of the array: ");
size = sc.nextInt();
System.out.println("Enter the elements: ");
for (int i = 0; i < size; i++) {
arr[i] = sc.nextInt();
if(arr[i]<0)
{throw new InputMismatchException("Invalid Input");}
else {
continue;
}
}
}

int getsum() {
if (size % 2 == 0) {
s = arr[0] + arr[size-1];
System.out.println("The no. of the elements is even.");
}
else{
s = arr[0]+arr[size-1]+arr[(size-1)/2];
System.out.println("The no. of the elements is odd.");
}
return s;
}

public static void main(String []args){
Arr_Sum_Len obj = new Arr_Sum_Len();
obj.getdata();
int sum = obj.getsum();
System.out.println("The sum of required elements: " + sum);
}
}

```

Output:

Array Size	Input	Expected Output	My Output
5	36,23,67,897,27	Size of array is odd Sum = 130	The no. Of elements is odd. The sum of required elements is 130
6	3,2,66,9,76,1,4	Size of array is even Sum = 4	The no of elements is even. The sum of required elements is 4
4	-6,6,-3,3,6	Exception	Exception

---

Q3: Demonstrate the 3x3 matrix multiplication using Arrays.

Program:

```

import java.util.Scanner;
import java.util.Arrays;
public class Matrix_Mult {
Scanner sc = new Scanner(System.in);
public int r1, r2, c1, c2;
int [][]mat1 = new int[100][100];
int [][]mat2 = new int[100][100];

```

```

int [][]prod = new int[100][100];
public
void getmat(){
    System.out.println("Enter the dimensions of M1: ");
    r1 = sc.nextInt();
    c1 = sc.nextInt();
    System.out.println("Enter the dimensions of M2: ");
    r2 = sc.nextInt();
    c2 = sc.nextInt();
    System.out.println("Enter the elements of M1");
    for(int i=0; i<r1; i++){
        for(int j=0; j<c1; j++){
            mat1[i][j]=sc.nextInt();
        }
    }
    System.out.println("Enter the elements of M"
+ "");
    for(int i=0; i<r2; i++){
        for(int j=0; j<c2; j++){
            mat2[i][j]=sc.nextInt();
        }
    }
}
void getprod(){

    for(int i=0; i<r1; i++){
        for(int j=0; j<c2; j++) {
            for (int k = 0; k < r2; k++) {
                prod[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}
void disp(){

    System.out.println("The resultant matrix is: ");
    for(int i=0; i<r1; i++){
        for(int j=0; j<c2; j++){
            System.out.print(prod[i][j]+ " ");
        }
        System.out.print("\n");
    }
}
public static void main(String []args){
    Scanner sc = new Scanner(System.in);
    Matrix_Mult obj = new Matrix_Mult();
    obj.getmat();
    obj.getprod();
    obj.disp();
}
}

```

Output:

Input	Expected Output	My Output
23,22,45   0,1,1	179, 158, 67	179, 158, 67
2,12,33   2,0,2	123, 101, 26	123, 101, 26
43,53,63   3,3,0	295, 232, 149	295, 232, 149

-2,22,45   32,54,-1	2297, 1377, -1406	2297, 1377, -1406
-23,12,33   -42,0,26	1169, -153, -1117	1169, -153, -1117
43,-53,68   73,33,-44	8566, 4566, -4413	8566, 4566, -4413

Bonus Question:

**He uses the small bricks to fill the gaps in the wall and the Larger ones to make the wall. He ran out of small bricks. Write a code that'll help him to break the larger bricks into smaller ones to fill the gaps in the walls.** (//Couldn't find the question so wrote what I remember)

Program:

```
import java.util.InputMismatchException;

import java.util.Scanner;

public class Bonus2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner scin = new Scanner(System.in);
        int brick1[] = new int[3];
        int brick2[] = new int[3];
        int vol1 = 1;
        int vol2 = 1;
        System.out.println("Enter Brick1 details");
        for(int i=0;i<3;i++) {
            brick1[i] = scin.nextInt();
            vol1 = vol1*brick1[i];
            if(brick1[i]<=0)
            {throw new InputMismatchException("Invalid Input, Value should be greater than 0");}
            else {
                continue;
            }
        }

        System.out.println("Enter Brick2 details");
        // System.out.println(vol1);
        for(int i=0;i<3;i++) {
            brick2[i] = scin.nextInt();
            vol2 = vol2*brick2[i];
            if(brick2[i]<=0)
            {throw new InputMismatchException("Invalid Input, Value should be greater than 0");}
            else {
                continue;
            }
        }
        // System.out.println(vol2);
        int dif = Math.abs(vol1 - vol2);
        System.out.println("Difference : " + dif);

    }

}
```

Output:

Volume of B1	Volume of B2	Difference	My Output
12,13,14	12,13,14	0	0
12,23,34	43,32,21	19512.0	19512
2,4,6	4,6,8	144.0	144
3,33,11	1,11,33	726.0	726

20,21,0	21,20,2	820	Exception
---------	---------	-----	-----------