**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**23MX26 - Java Programming Laboratory**

**Hands-on Worksheet 3**

Write a program to accept 'n' (taken from test data) numbers of elements using dynamic memory allocation and then print the element at position 'p' (taken from test data) after sorting the elements in ascending order.

For example:

If the value of n is 5 and value of p is 3 then the program will accept five numbers as input from the test data, sort them in ascending order and display the third number as output.

If the input is 20,30,90,40,80 then it will display 40 as output.

Sample Test Cases

|  | **Input** | **Output** |
| --- | --- | --- |
| Test Case 1 | 5  3  50  40  10  20  30 | The desired element is 30 |
| Test Case 2 | 7  2  40  -70  20  5  3  50  30 | The desired element is 3 |
| Test Case 3 | 4  2  7  2  9  5 | The desired element is 5 |
| Test Case 4 | 6  4  9  10  4  30  40  5 | The desired element is 10 |

2. Write a program which dynamically allocates two one-dimensional arrays of 'n' elements each (supplied from test data) and store their sum in a separate array and print the third array.

| **Private Test cases used for evaluation** | **Input** | **Expected Output** |
| --- | --- | --- |
| Test Case 1 | 5  6  10  -4  6  -5  2  3  4  -5  7 | 8\n  13\n  0\n  1\n  2\n |
| Test Case 2 | 7  -9  -6  -4  -3  -1  -2  -8  9  6  4  3  1  2  8 | 0\n  0\n  0\n  0\n  0\n  0\n  0\n |

3. Write a program to find the total sum of the diagonal elements of a square matrix.

Sample Test Cases

|  | **Input** | **Output** |
| --- | --- | --- |
| Test Case 1 | 3  4  4  4  5  5  5  6  6  6 | Sum of the diagonal elements is = 30 |
| Test Case 2 | 5  3  4  1  2  6  8  6  0  1  -1  -4  -7  -8  6  4  2  7  0  1  2  3  4  7  8  9 | Sum of the diagonal elements is = 20 |
| Test Case 3 | 4  1  2  3  4  1  2  3  4  1  2  3  4  1  2  3  4 | Sum of the diagonal elements is = 20 |

4. Write a program to find Sum of series S=1+(1+2)+(1+2+3)+..….+(1+2+3+…+n)

| **Private Test cases used for evaluation** | **Input** | **Expected Output** |
| --- | --- | --- |
| Test Case 1 | 10 | The Sum of Series up to Value 10 = 220\n |
| Test Case 2 | 20 | The Sum of Series up to Value 20 = 1540\n |

5. **Complete the Java code which takes (i) size of the array (number of array elements) and (ii) the array elements as input and then counts and displays (i) the number of elements that are repeated and (ii) the repeated elements (in order of their occurrence in the input array)**

For example if the size of the array is 10 and the elements are

10, 30, 20, 45, 60, 20, 20, 60, 20, 30

The output will be:

Total Repeated elements = 3   
Repeated elements are: 30 20 60

Sample Test Cases

|  | **Input** | **Output** |
| --- | --- | --- |
| Test Case 1 | 8  20  40  50  40  60  60  70  90 | Total Repeated elements = 2  Repeated elements are: 40 60 |
| Test Case 2 | 8  20  40  50  40  60  60  70  90 | Total repeated elements = 2  Repeated elements are: 40 60 |
| Test Case 3 | 10  10  30  20  45  60  20  20  60  20  30 | Total Repeated elements = 3  Repeated elements are: 30 20 60 |

6. Complete the Java program which takes size of the array and array elements as input and puts the prime and composite elements of the array in two separate arrays (according to their occurrence in the input array) .  
For example:  
If size of an array is 10 and the elements are 3, 9, 60, 5, 17, 40, 70, 18, 100 and 43 then the output will be  
  
Elements of Prime array: 3 5 17 43  
Elements of Composite array: 9 60 40 70 18 100  
  
Use the following print statements  
  
printf("Elements of Prime array: ");  
printArray(Prime, PrimeCount);  
printf("\nElements of Composite array: ");  
printArray(Composite, CompositeCount);

Sample Test Cases

|  | **Input** | **Output** |
| --- | --- | --- |
| Test Case 1 | 10  6  78  23  49  56  31  51  76  6  5 | Elements of Prime array: 23 31 5  Elements of Composite array: 6 78 49 56 51 76 6 |
| Test Case 2 | 10  3  9  60  5  17  40  70  18  100  43 | Elements of Prime array: 3 5 17 43  Elements of Composite array: 9 60 40 70 18 100 |