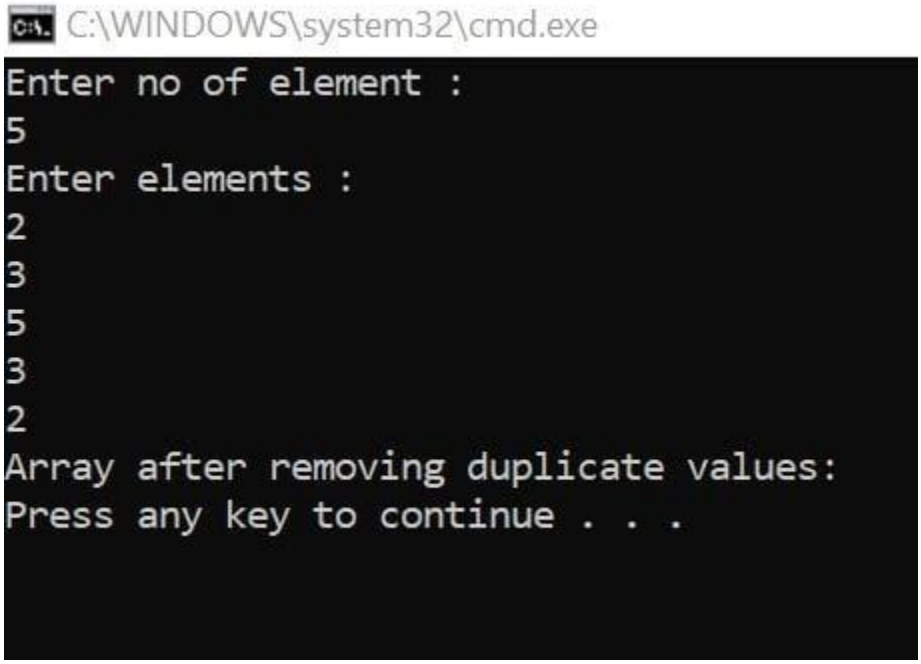
Practical-3: Array & User Defined Function

1. Program to remove duplicate elements from array**Code :**

```
class P3_1
    {
        public static void run()
        {
            //Remove duplicate from array.
            Console.WriteLine("Enter no of element : ");
            int size = Convert.ToInt32(Console.ReadLine());

            int[] arr = new int[size];
            Console.WriteLine("Enter elements : ");

            for(int i=0;i < size; i++)
            {
                arr[i] = Convert.ToInt32(Console.ReadLine());
            }

            for(int i=0; i < size; ++i)
            {
                for(int j=i+1; j < size; ++j)
                {
                    if(arr[j] == arr[i])
                    {
                        for(int k = j; k < size-1; ++k)
                        {
                            arr[k] = arr[k + 1];
                        }
                        size--;
                        j--;
                    }
                }
            }
```

```
            // using in built function
            // int[] uni = arr.Distinct().ToArray();

            Console.WriteLine("Array after removing duplicate values: ");

        for (int i = 0; i < size; i++)
        {
            Console.Write(" "+arr[i]);
        }
    }
}
```

Output:



2. Multiplication of 2-dimensional matrix using 2-d array.

Code :

```
class P3_2
    {
        public static void run()
        {
            // product of 2x2 matrix.
```

```
            Console.WriteLine("Enter number of rows : ");
            int r = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter number of column : ");
            int c = Convert.ToInt32(Console.ReadLine());


            int[,] m1 = new int[r,c];
            int[,] m2 = new int[r, c];
            int[,] res = new int[r, c];

            Console.WriteLine("Enter elements in 1st matrix : ");
            for (int i = 0; i < r; ++i)
            {
                for(int j=0; j < c; ++j)
                {
                    m1[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            Console.WriteLine("Enter elements in 2st matrix : ");
            for (int i = 0; i < r; ++i)
            {
                for (int j = 0; j < c; ++j)
                {
                    m2[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            for(int i=0; i< r; ++i)
            {
                for(int j=0;j < c; ++j)
                {
                    res[i, j] = 0;
                    for(int k =0; k < c; k++)
                    {
                        res[i, j] += m1[i, k] * m2[k, j];
```

```
                }
                Console.Write(res[i, j] + "\t");
            }
            Console.WriteLine();
        }

        /*for(int i =0;i<r; ++i)
        {
            for(int j=0; j< c; ++j)
            {
                Console.Write(res[i, j]+"\t");
            }
            Console.WriteLine();
        }*/
    }
        }
```

Output :

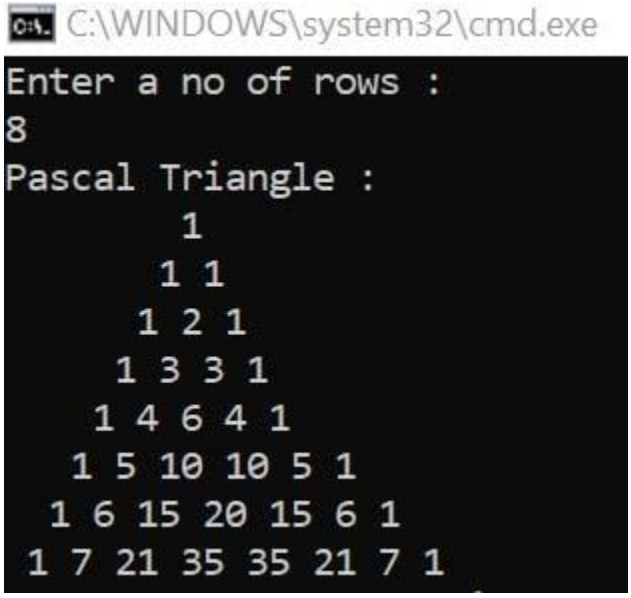
```
C:\WINDOWS\system32\cmd.exe
Enter number of rows :
2
Enter number of column :
2
Enter elements in 1st matrix :
2
2
2
2
Enter elements in 2st matrix :
2
2
2
2
8       8
8       8
```

3. Generate Pascal Triangle using jagged array.

Code :

```
class P3_3
    {
        public static void run()
        {
            // pascal traingle using jagged array
            Console.WriteLine("Enter a no of rows : ");
            int rows = int.Parse(Console.ReadLine());
            int[,] arr = new int[rows,rows];
            Console.WriteLine("Pascal Triangle : ");

            for (int line = 0; line < rows; line++)
            {
                for(int k =0; k < rows - line; k++)
                {
                    Console.Write(" ");
                }
                for (int i = 0; i <= line; i++)
                {
                    if (line == i || i == 0)
                        arr[line, i] = 1;
                    else
                         arr[line, i] = arr[line - 1, i - 1] + arr[line -
1, i];

                     Console.Write(arr[line, i]+" ");
                }
                Console.WriteLine("");
        }
        }
    }
```

Output :



4. User define function to sort an array.

Code:

```
class P3_4
    {
        public static void run()
        {
            //Sorting of array.


          Console.WriteLine("Enter a size of an array : ");
        int size = int.Parse(Console.ReadLine());
        int[] arr = new int[size];
        for(int i=0; i<size; ++i)
        {
            arr[i] = int.Parse(Console.ReadLine());
        }


        Sort(arr, size);
        Console.WriteLine("Sorted array is : ");
        for (int i = 0; i < size; ++i)
        {
```

```
            Console.WriteLine( arr[i] +" ");
        }
    }


    static void Sort(int[] arr,int n)
    {
        for(int i=0; i<n; ++i)
        {
            for(int j = i+1; j<n; ++j)
            {
                if(arr[j] < arr[i])
                {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}
```

Output:

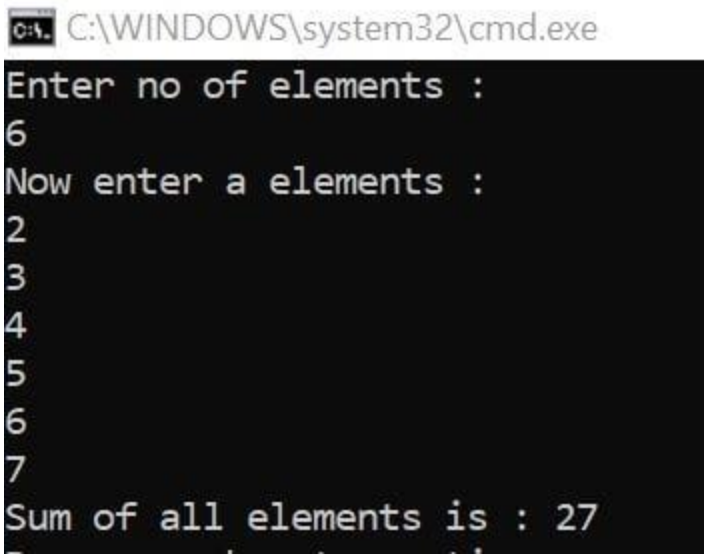
```
C:\WINDOWS\system32\cmd.exe
Enter a size of an array :
7
33
55
23
43
54
43
-3
Sorted array is :
-3
23
33
43
43
54
55
```

5. Use of **Params** keyword in program.

Code :

```
class P3_5
    {
        public static void run()
        {
            // dispaly use of params keywords using program

            Console.WriteLine("Enter no of elements : ");
            int size = int.Parse(Console.ReadLine());
            Console.WriteLine("Now enter a elements : ");
            int[] no = new int[size];
            for(int i = 0; i < size; i++)
            {
                no[i] = int.Parse(Console.ReadLine());
            }
            Console.WriteLine("Sum of all elements is : "+ find_sum(no));
        }
        static int find_sum(params int[] n)
        {
            int sum = 0;
        foreach(int i in n){
            sum += i;
        }
        return sum;
    }
}
```

Output :



6. Discuss **Out** and **Ref** Keywords with the help of a program.

Code :

```
class P3_6
    {
        public static void run()
        {
            // discuss out and ref using program.


            // for REF Keyword.
            int num = 10;
            ref_ass(ref num);
            Console.WriteLine(num);


            //for Out Keyword.
            int val;
            out_ass(out val);
            Console.WriteLine(val);
    }
    static void ref_ass(ref int i)
    {
        //for ref keywords
        i++;
```

```
    }
    static void out_ass(out int j)
    {
        //for out keywords
        j = 100;
        j++;
    }
}
```

Output :

```
Which question you wnat :
6
==============>>> Program - 6 running ....
11
101
```