Assignment 5

1. Create following types of arrays
   1. Integer
   2. String

Use System.Array class to perform following operations on them

Copy, Sort, Clear, Reverse

Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SystemArray\_Program

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter size of an array");

int size\_of\_arr = int.Parse(Console.ReadLine());

int[] array = new int[size\_of\_arr];

string[] string\_array = new string[size\_of\_arr];

Console.WriteLine("Enter int type element in array\n");

for (int i = 0; i < size\_of\_arr; i++)

{

array[i] = int.Parse(Console.ReadLine());

}

Console.WriteLine("\nOutput before applying any method \n");

foreach (int val in array)

{

Console.Write(val + " ");

}

Console.WriteLine("\n");

int[] array2 = new int[size\_of\_arr];

Array.Copy(array, array2, size\_of\_arr);//Copy() Method

Console.WriteLine("After copying one array in array2\n");

foreach (int val in array2)

{

Console.Write(val + " ");

}

Console.WriteLine("\nafter clearing array2\n");

Array.Clear(array2, 0, size\_of\_arr);//clear method to clear array2

foreach(int val in array2)

{

Console.Write(val + " ");

}

Console.WriteLine();

Array.Reverse(array);//After reversing array

foreach (int val in array)

{

Console.Write(val + " ");

}

Console.WriteLine("\n");

Array.Sort(array);//sorting array

Console.WriteLine("\nAfter Sorting array\n");

foreach (int val in array)

{

Console.Write(val + " ");

}

Console.WriteLine("\nInsert string in string array\n");

for (int i = 0; i < size\_of\_arr; i++)

{

string\_array[i] = Console.ReadLine();

}

Console.WriteLine("Output of String\_array before appplying any method\n");

foreach (string ele in string\_array)

{

Console.Write(ele + " ");

}

Console.WriteLine("\n");

Array.Reverse(string\_array);

Console.WriteLine("After Reversing string array\n");

foreach (string ele in string\_array)

{

Console.Write(ele + " ");

}

Console.WriteLine("\n");

Array.Sort(string\_array);

Console.WriteLine("\nAfter sorting string\_array\n");

foreach (string ele in string\_array)

{

Console.Write(ele + " ");

}

Console.WriteLine("\n");

string[] array\_string2 = new string[size\_of\_arr];

Array.Copy(string\_array, array\_string2, size\_of\_arr);

Console.WriteLine("\nAfter copying array\_string into array\_string2\n");

foreach(string ele in array\_string2)

{

Console.Write(ele + " ");

}

Console.WriteLine("\nafter clearing\n");

Array.Clear(string\_array, 0, size\_of\_arr);

foreach (string ele in string\_array)

{

Console.Write(ele + " ");

}

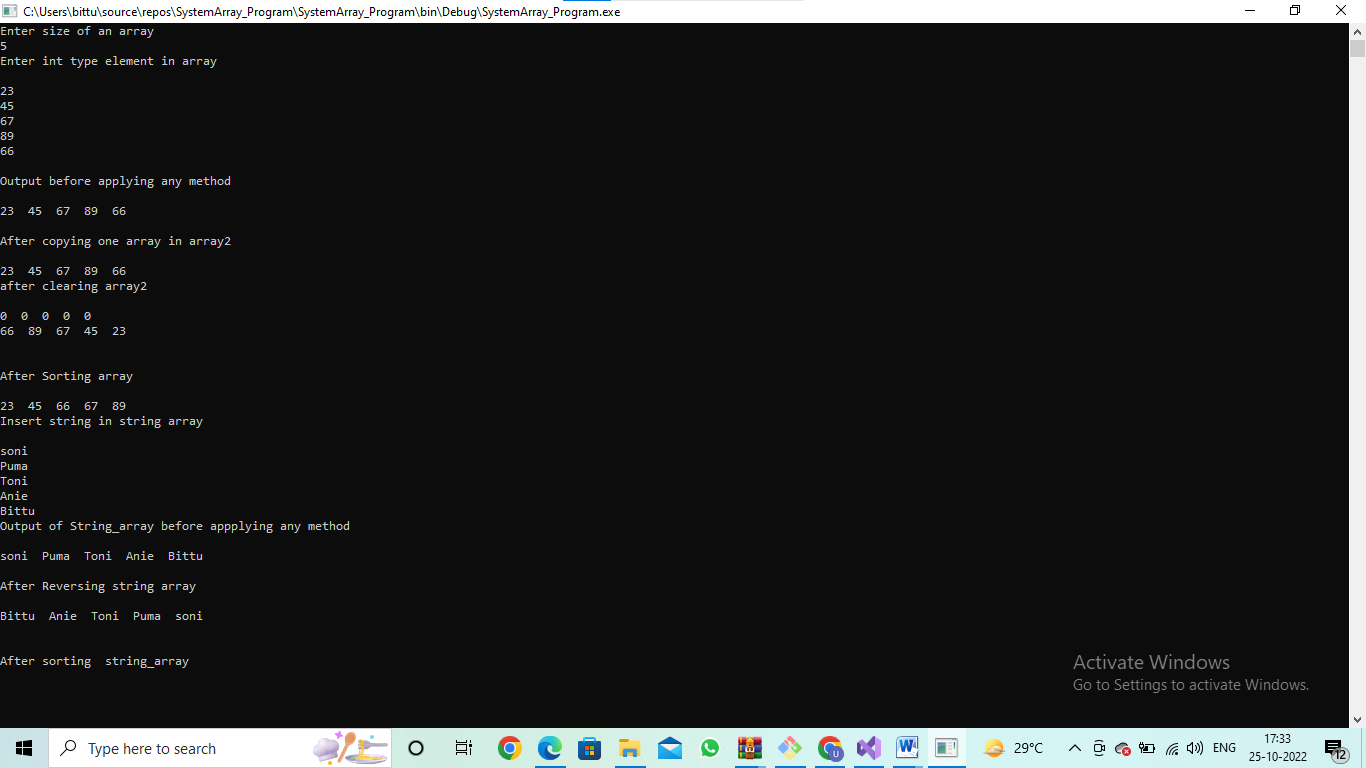
Console.ReadLine();

}

}

}

OUTPUT



Accept input from user through Console.

1. Use collection class such as ArrayList to hold more than one employee objects in Employee Management application. Display all Employee details which are stored in collection.

Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace ArrayList1

{

public class Employee

{

string name;

int emp\_id;

double salary;

public Employee(string name, int emp\_id, double salary)

{

this.name = name;

this.emp\_id = emp\_id;

this.salary = salary;

}

public override string ToString()

{

return

String.Format("{0,-10} emp\_id : {1,6} salary : {2}",name,emp\_id,salary);

}

}

internal class Program

{

static void Main(string[] args)

{

ArrayList emp = new ArrayList();

Console.WriteLine("Add element to the list");

emp.Add(new Employee("Dipanshu", 012, 23450.89));

emp.Add(new Employee("Vidya", 034, 4500.78));

emp.Add(new Employee("Sonakshi", 098, 34560.89));

emp.Add(new Employee("Samiksha", 078, 34000.89));

Console.WriteLine("Employee Details\n");

foreach(Employee empl in emp)

{

Console.WriteLine(" " + empl);

}

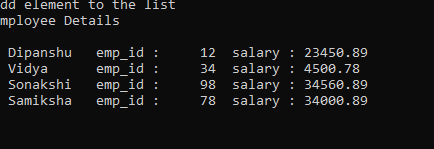
Console.ReadLine();

}

}

}

Output



3.Write a console based program to create a linked list of Employee objects using the generic class List<>.Perform following operations on the list:

1. Add a new employee
2. Display the list of employees.
3. Total number of employees in the list

Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment5quet3LinkedList

{

internal class Program

{

static void Main(string[] args)

{

LinkedList<string> employee = new LinkedList<string>();

//Adding a new employee

employee.AddFirst("Sandhya");

employee.AddFirst("Anjali");

employee.AddFirst("Radha");

Console.WriteLine("Display the list of an employee\n");

foreach(string emp in employee)

{

Console.WriteLine(emp+ "\n");

}

Console.WriteLine("Total number of employee in the list : " + employee.Count());

Console.WriteLine("\n");

//Searching employee name is it present or not Assignment 5 quetion 5

Console.WriteLine("The employee name Anjali is present in linked list : " +

employee.Contains("Anjali"));

Console.WriteLine("\n");

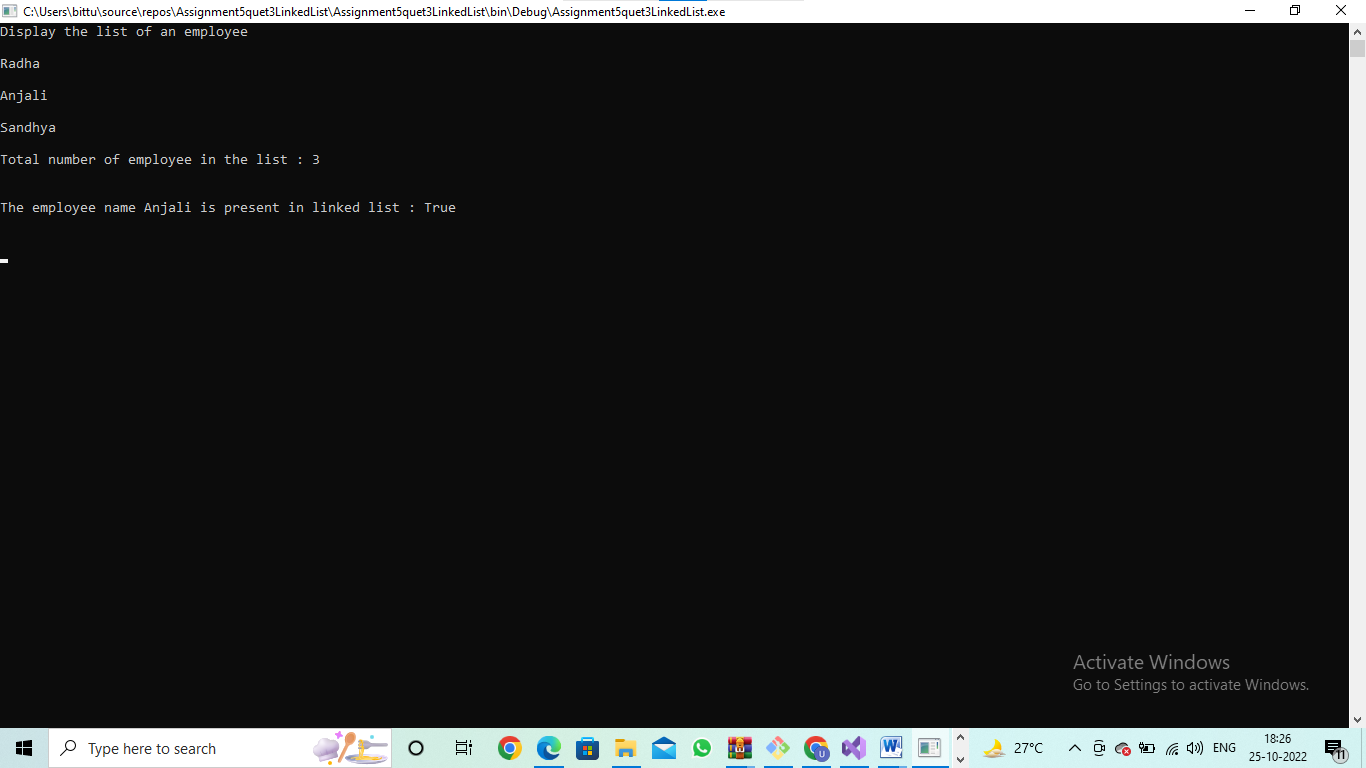
Console.ReadLine();

}

}

}

OUTPUT



1. Write Custom Generic class MyStack based on assignment of previous session, with

Push() and Pop() methods to store any kind of .NET Type.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

using static System.Console;

namespace MyStackStoreAnyType

{

internal class Program

{

static void Main(string[] args)

{

Stack st = new Stack();

st.Push("Shubham");

st.Push("banana");

st.Push(980.89);

st.Push(45);

st.Push('a');

st.Push(345.67f);

st.Push(234578.90D);

foreach(object obj in st)

{

WriteLine(obj);

}

WriteLine("\n");

WriteLine("Popping the element from stack {0}",st.Pop());

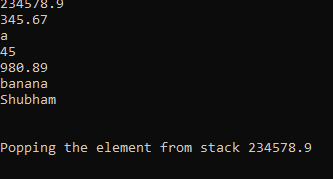
ReadLine();

}

}

}

Output



## Now try these to get a complete grip…

1. In the assignment 3 above, add a functionality to search an employee on name in the List<>.