}//V main Print List in Method

Console.Write($"Shot targets: {countshoottargets} -> ");

PrintListElements(SequenceTargets);

}

static void PrintListElements(List<int> Elements)

{

foreach (int Number in Elements)

{

Console.Write(Number + " ");

}

}

List<string> Lessons = Console.ReadLine()

.Split(", ")

.ToList();

string command = " ";

while ((command = Console.ReadLine()) != "course start")

{

string[] Arr1 = command.Split(':').ToArray();

string typecommand = Arr1[0];

Valid, Invalid Index

string entered = Console.ReadLine();

List<string> allNumbers =entered.Split(" ").ToList();

string command = " ";

while ((command = Console.ReadLine()) != "End")

{

string[] Arr1 = command.Split(' ').ToArray();

string Operation = Arr1[0];

switch (Operation)

{

case "Add":

allNumbers.Add(Arr1[1]);

break;

case "Insert":

int index1 = int.Parse(Arr1[2]);

if (index1 < 0 || index1 > (allNumbers.Count - 1))

{

Console.WriteLine("Invalid index");

}

else

{

allNumbers.Insert(int.Parse(Arr1[2]), Arr1[1]);

}

if (allgroseries.Contains(Arr1[1]))

{

string oldValue = Arr1[1];

string newValue = Arr1[2];

int index = allgroseries.IndexOf(oldValue);

allgroseries[index] = newValue;

}

else

{

continue;

}

int lessindex3 = Lessons.FindIndex(a => a.Contains(Arr1[1]));

case "Swap":

bool exist1 = Lessons.Contains(Arr1[1]);

bool exist2 = Lessons.Contains(Arr1[2]);

if (exist1 == true && exist2 == true)

{

string firstless = Arr1[1];

string secondless = Arr1[2];

int firslessindex = Lessons.FindIndex(a => a.Contains(Arr1[1]));

int secondindex = Lessons.FindIndex(a => a.Contains(Arr1[2]));

string firstex = firstless + "-Exercise";

string secondex = secondless + "-Exercise";

Lessons.RemoveAt(firslessindex);

Lessons.Insert(firslessindex, secondless);

Lessons.RemoveAt(secondindex);

Lessons.Insert(secondindex, firstless);

if (Lessons.Contains(secondex))

{

Lessons.Remove(secondex);

Lessons.Insert(firslessindex + 1, secondex);

}

if (Lessons.Contains(firstex))

{

// Lessons.RemoveAt(secondindex + 1);

Lessons.Remove(firstex);

Lessons.Insert(secondindex + 1, firstex);

}

}

break;

case "Exercise":

string Exersise1 = " ";

if (!Lessons.Contains(Arr1[1]))

{

Exersise1 = Arr1[1] + "-Exercise";

Lessons.Add(Arr1[1]);

Lessons.Add(Exersise1);

}

else

{

Exersise1 = Arr1[1] + "-Exercise";

int lessindex3 = Lessons.FindIndex(a => a.Contains(Arr1[1]));

Lessons.Insert(lessindex3+1, Exersise1);

}

break;

1. //for (int i = 0; i < numbers.Count; i++)
2. //{
3. // if (numbers[i] < 0)
4. // {
5. // numbers.Remove(numbers[i]);
6. // i--;
7. // }
8. //}
9. numbers.RemoveAll(x => x < 0);
11. //Error
12. //foreach (var number in numbers)
13. //{
14. // if (number < 0)
15. // {
16. // numbers.Remove(number);
17. // }
18. //}
20. numbers.Reverse();

RotatonArray

|  |
| --- |
| using System; |
|  | using System.Linq; |
|  |  |
|  | namespace P04.ArrayRotations |
|  | { |
|  | internal class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | int[] arr = Console.ReadLine() |
|  | .Split(" ", StringSplitOptions.RemoveEmptyEntries) |
|  | .Select(int.Parse) |
|  | .ToArray(); |
|  | int rotationsCount = int.Parse(Console.ReadLine()); |
|  |  |
|  | int timesToRotate = rotationsCount % arr.Length; |
|  | for (int r = 1; r <= timesToRotate; r++) |
|  | { |
|  | int firstEl = arr[0]; |
|  | for (int i = 1; i < arr.Length; i++) |
|  | { |
|  | arr[i - 1] = arr[i]; |
|  | } |
|  | arr[arr.Length - 1] = firstEl; |
|  | } |
|  |  |
|  | Console.WriteLine(String.Join(" ", arr)); |
|  | } |
|  | } |
|  | } |

3 ZigZagArrays

|  |  |
| --- | --- |
| using System; |  |
|  |  | using System.Linq; |
|  |  |  |
|  |  | namespace P03.ZigZagArrays |
|  |  | { |
|  |  | internal class Program |
|  |  | { |
|  |  | static void Main(string[] args) |
|  |  | { |
|  |  | int n = int.Parse(Console.ReadLine()); |
|  |  |  |
|  |  | int[] firstArray = new int[n]; |
|  |  | int[] secondArray = new int[n]; |
|  |  | for (int i = 1; i <= n; i++) |
|  |  | { |
|  |  | //[1, 4] |
|  |  | int[] numbers = Console.ReadLine() |
|  |  | .Split(" ") |
|  |  | .Select(int.Parse) |
|  |  | .ToArray(); |
|  |  | int firstNum = numbers[0]; |
|  |  | int secondNum = numbers[1]; |
|  |  |  |
|  |  | if (i % 2 != 0) |
|  |  | { |
|  |  | //Odd -> 1, 3, 5, 7... |
|  |  | firstArray[i - 1] = firstNum; |
|  |  | secondArray[i - 1] = secondNum; |
|  |  | } |
|  |  | else |
|  |  | { |
|  |  | //Even -> 2, 4, 6, 8... |
|  |  | firstArray[i - 1] = secondNum; |
|  |  | secondArray[i - 1] = firstNum; |
|  |  | } |
|  |  | } |
|  |  |  |
|  |  | Console.WriteLine(String.Join(" ", firstArray)); |
|  |  | Console.WriteLine(String.Join(" ", secondArray)); |
|  |  | } |
|  |  | } |
|  |  | } |