12214994_SaurabhRana

Title

Sorting Strings by Length in Descending Order

Question No : 1 / 1

Problem Statement

You need to develop a program in **C#** that sorts a list of strings based on their lengths in **descending order**.
If two strings have the same length, they should **maintain their relative order** as in the original list.

The program should be written within the **Program** class in the **Program.cs** file.

All the classes must be in public

Write C# program that should

1. Allow the user to input a series of strings dynamically separated by spaces
2. Sort the strings according to their lengths in descending order
3. Output the sorted list of strings

Example:
Input: `apple banana pear grape`
Output: `banana apple grape pear`

Input Format

- A single line containing space-separated strings

Output Format

- A single line containing the sorted strings, sorted primarily by length in descending order, separated by spaces

Sample Input 1

```
banana pear grape apple
```

Sample Output 1

```
banana grape apple pear
```

Sample Input 2

```
tiger elephant dog lion cat
```

Sample Output 2

```
elephant tiger lion dog cat
```

Test Cases

Test Case 1

Input

```
apple banana pear grape
```

Expected Output

```
banana apple grape pear
```

Test Case 2

Input

```
one three five seven
```

Expected Output

```
three seven one five
```

Test Case 3

Input

```
a bb ccc dddd
```

Expected Output

```
dddd ccc bb a
```

Test Case 4

Input

```
same size test case
```

Expected Output

```
same size test case
```

(All strings have equal length, original order preserved)

Test Case 5

Input

```
hello hi h welcome
```

Expected Output

```
welcome hello hi h
```

Commands to Run the Project

```
cd dotnetapp
dotnet run
dotnet build
dotnet clean
```

Note

- Sorting must be **stable** (relative order preserved for same-length strings)
- Input is space-separated
- Case sensitivity is preserved

```
public class Program

{

   public static void Main()

   {



      string input = Console.ReadLine()??"";



      string[] words = input.Split(' ', StringSplitOptions.RemoveEmptyEntries);



      var sortedWords = words.Select((word, index) => new { word, index }).OrderByDescending(x => x.word.Length).ThenBy(x => x.index).Select(x => x.word);



      Console.WriteLine(string.Join(" ", sortedWords));

   }

}
```