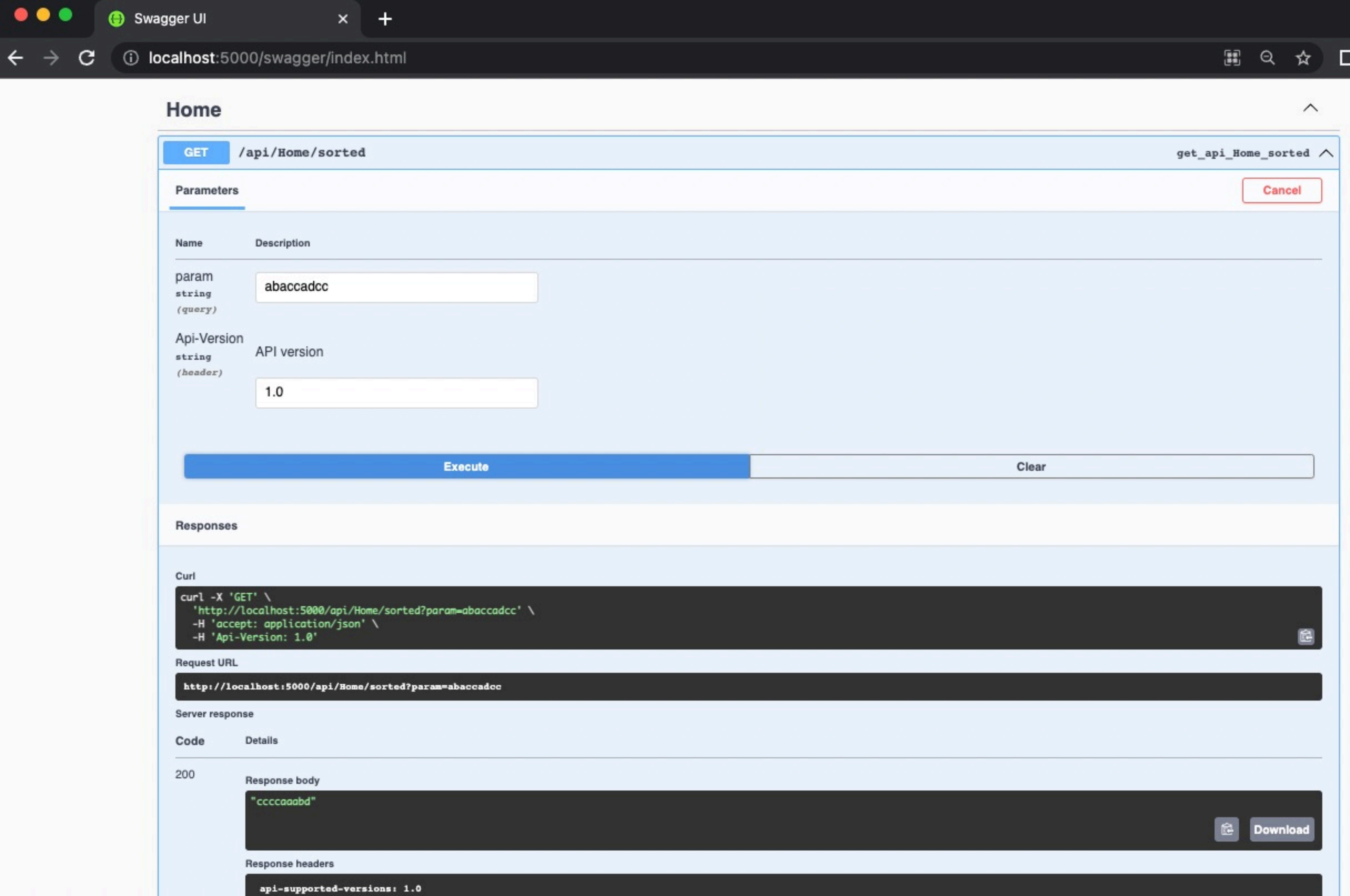

Swagger UI

localhost:5000/swagger/index.html

Home

GET /api/Home/sorted get_api_Home_sorted

Parameters Cancel

Name	Description
param string (query)	abaccadcc
Api-Version string (header)	API version 1.0

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5000/api/Home/sorted?param=abaccadcc' \
  -H 'accept: application/json' \
  -H 'Api-Version: 1.0'
```

Request URL

```
http://localhost:5000/api/Home/sorted?param=abaccadcc
```

Server response

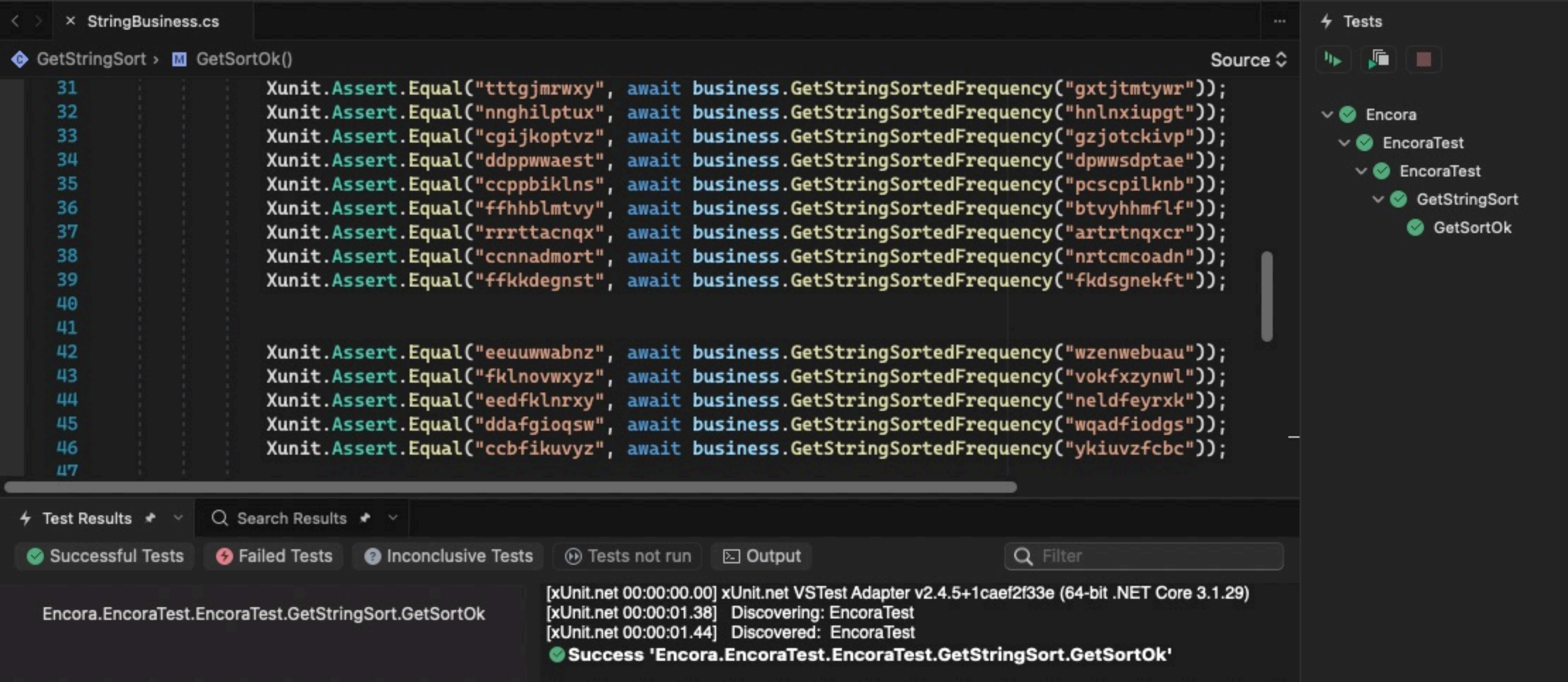
Code	Details
200	Response body

```
"ccccaaabd"
```

Download

Response headers

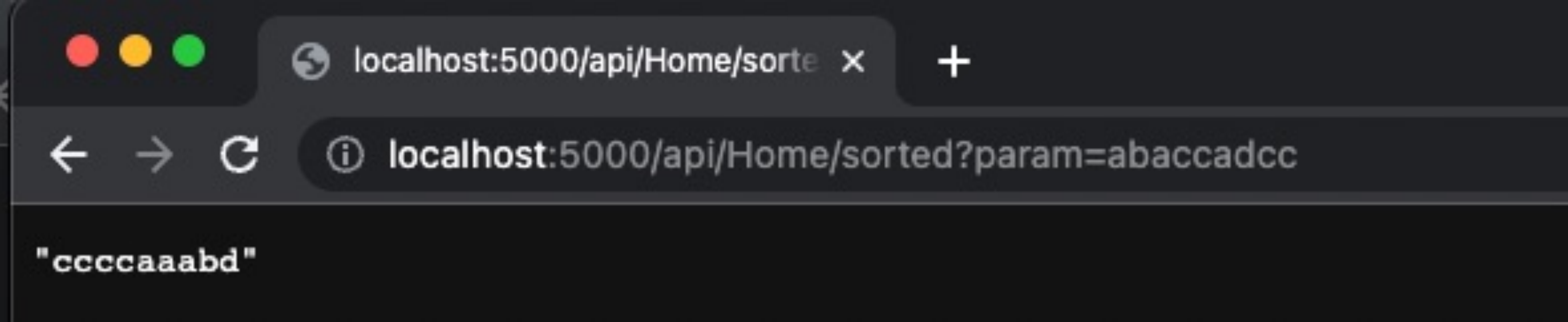
```
api-supported-versions: 1.0
```



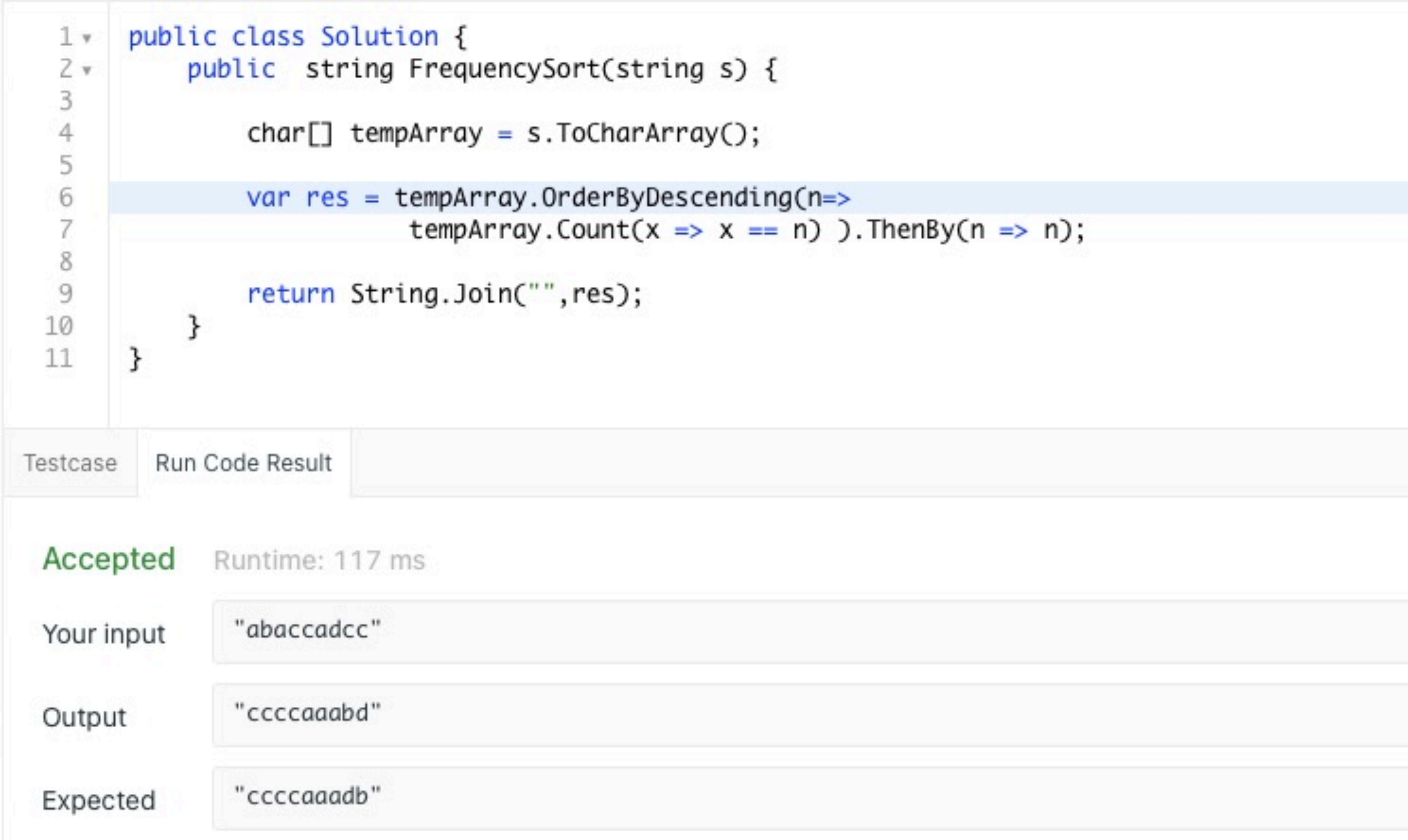
```
                Xunit.Assert.Equal("tttgjmrwxy", await business.GetStringSortedFrequency("gxtjtmtywr"));
                Xunit.Assert.Equal("nnghilptux", await business.GetStringSortedFrequency("hnlnxiupgt"));
                Xunit.Assert.Equal("cgijkoptvz", await business.GetStringSortedFrequency("gzjotckivp"));
                Xunit.Assert.Equal("ddppwwaest", await business.GetStringSortedFrequency("dpwwsdptae"));
                Xunit.Assert.Equal("ccppbiklns", await business.GetStringSortedFrequency("pcscpilknb"));
                Xunit.Assert.Equal("ffhhblmtvy", await business.GetStringSortedFrequency("btvyhhmflf"));
                Xunit.Assert.Equal("rrrttacnqx", await business.GetStringSortedFrequency("artrtnqxcr"));
                Xunit.Assert.Equal("ccnnadmort", await business.GetStringSortedFrequency("nrtcmcoadn"));
                Xunit.Assert.Equal("ffkkdegnst", await business.GetStringSortedFrequency("fkdsgnekft"));


                Xunit.Assert.Equal("eeuuwwabnz", await business.GetStringSortedFrequency("wzenwebuau"));
                Xunit.Assert.Equal("fklnovwxyz", await business.GetStringSortedFrequency("vokfxzynwl"));
                Xunit.Assert.Equal("eedfklnrxy", await business.GetStringSortedFrequency("neldfeyrxk"));
                Xunit.Assert.Equal("ddafgioqsw", await business.GetStringSortedFrequency("wqadfiodgs"));
                Xunit.Assert.Equal("ccbfikuvyz", await business.GetStringSortedFrequency("ykiuvzfcbc"));
```

Encora.EncoraTest.EncoraTest.GetStringSort.GetSortOk

```
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.4.5+1caef2f33e (64-bit .NET Core 3.1.29)
[xUnit.net 00:00:01.38]   Discovering: EncoraTest
[xUnit.net 00:00:01.44]   Discovered:  EncoraTest
Success 'Encora.EncoraTest.EncoraTest.GetStringSort.GetSortOk'
```


localhost:5000/api/Home/sorte
localhost:5000/api/Home/sorted?param=abaccadcc

```
"ccccaaabd"
```



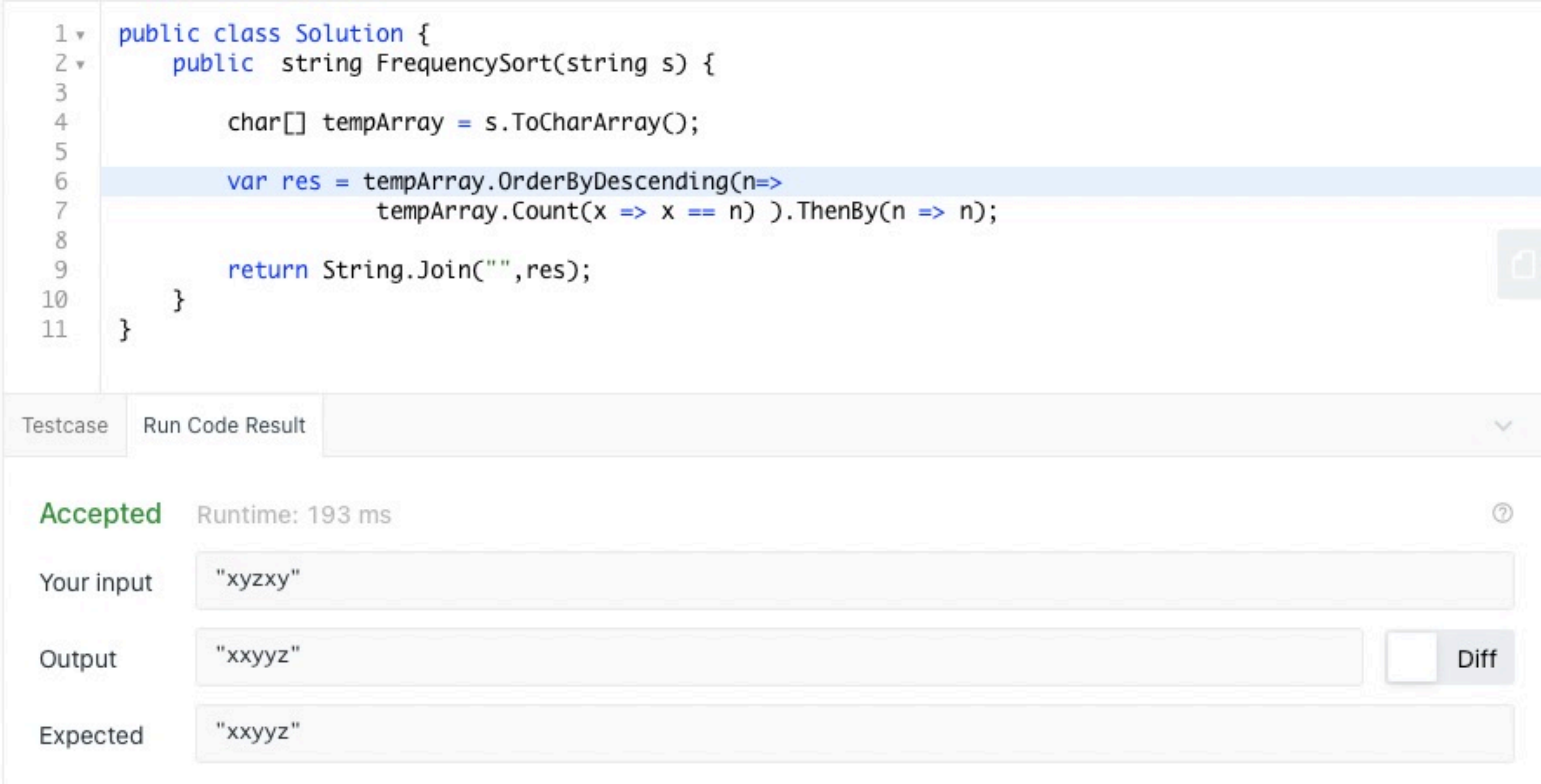
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                  tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 117 ms

Your input "abaccadcc"

Output "ccccaaabd"

Expected "ccccaaadb"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

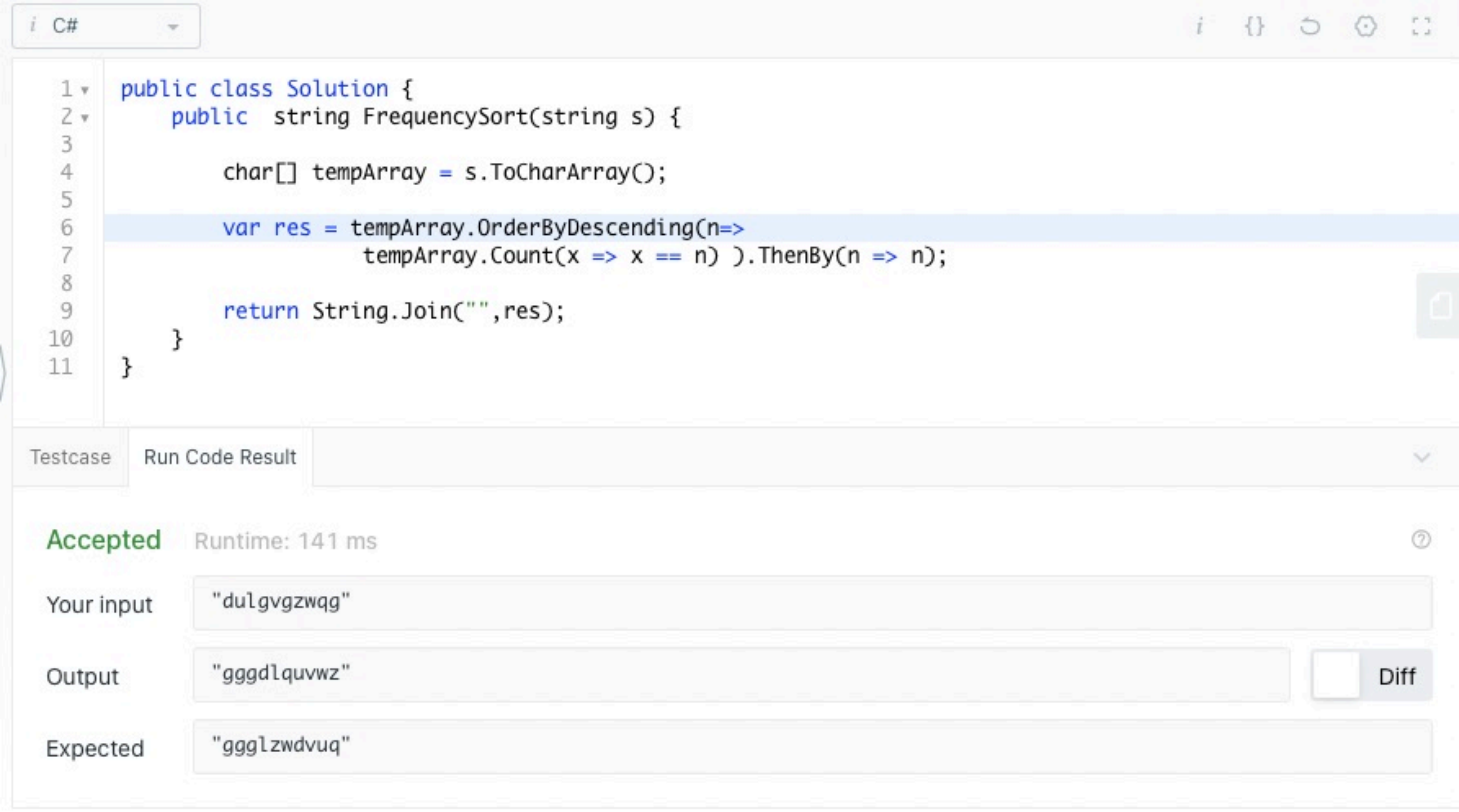
Testcase | Run Code Result

Accepted Runtime: 193 ms

Your input "xyzxy"

Output "xxyyz" Diff

Expected "xxyyz"



C#

```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

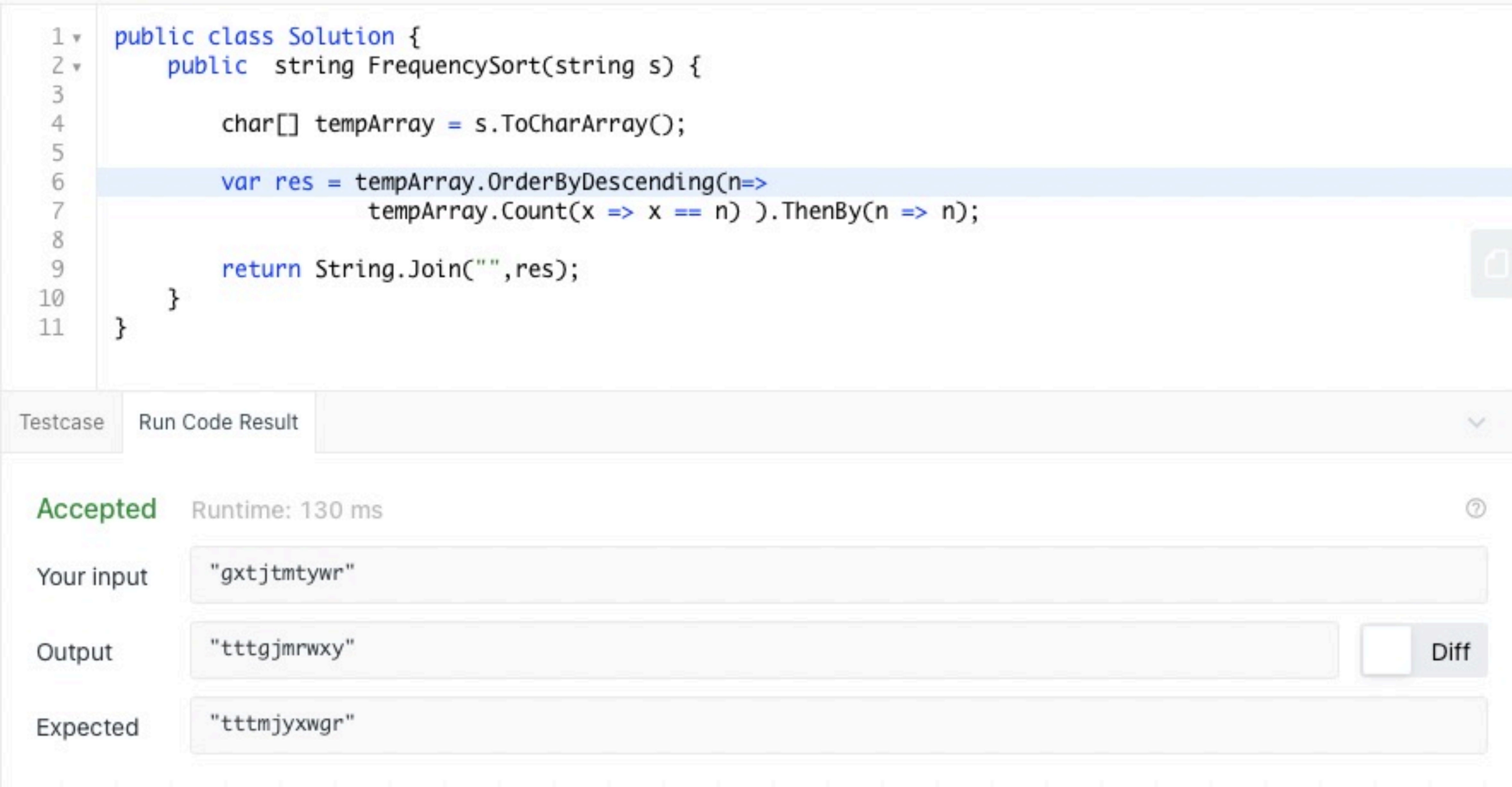
Testcase | Run Code Result

Accepted Runtime: 141 ms

Your input "dulgvgzwqg"

Output "gggdlquvwz" Diff

Expected "ggglzwdvuq"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

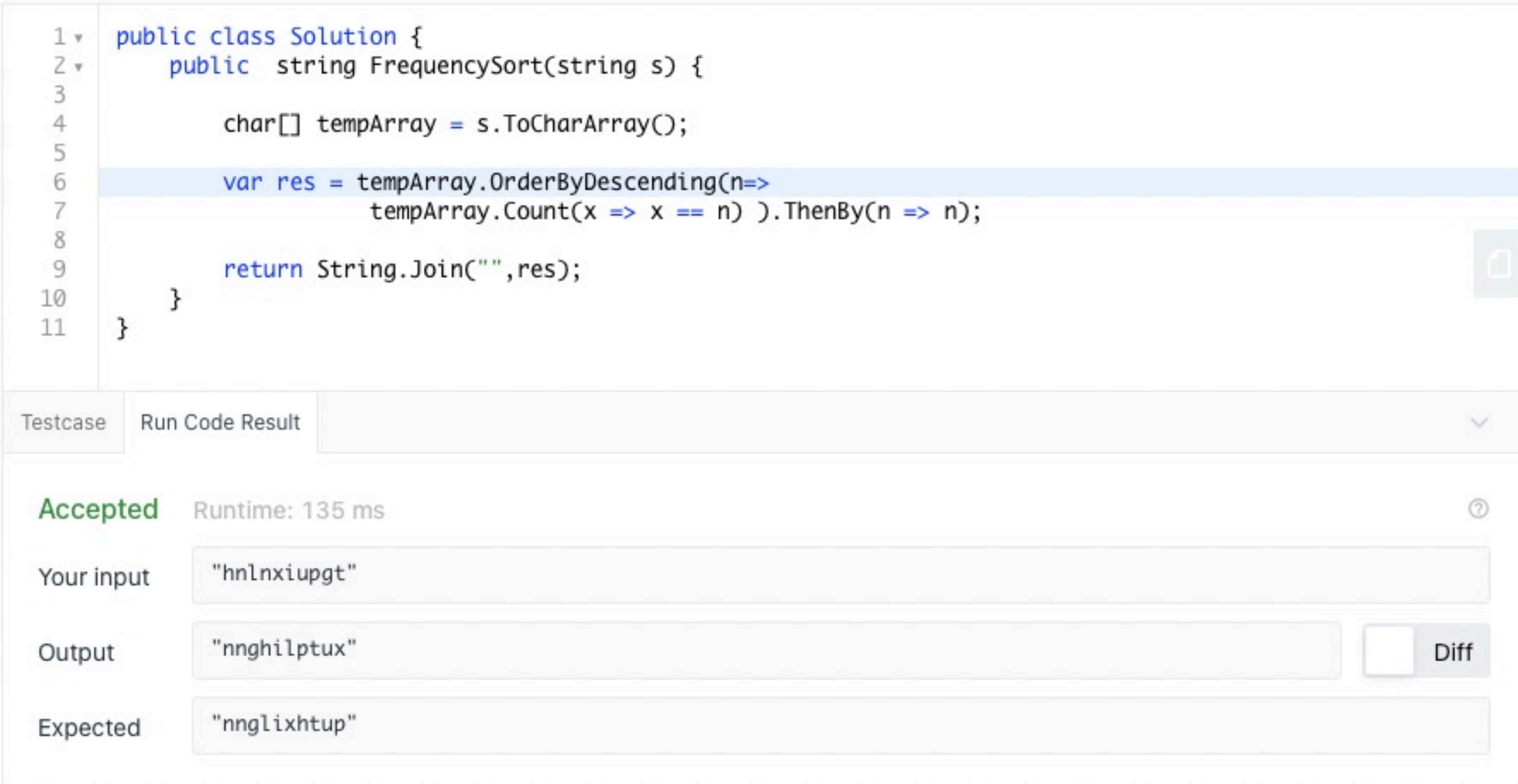
Testcase | Run Code Result

Accepted Runtime: 130 ms

Your input "gxtjtmtywr"

Output "tttgjmrwxy"

Expected "tttmjyxwgr"



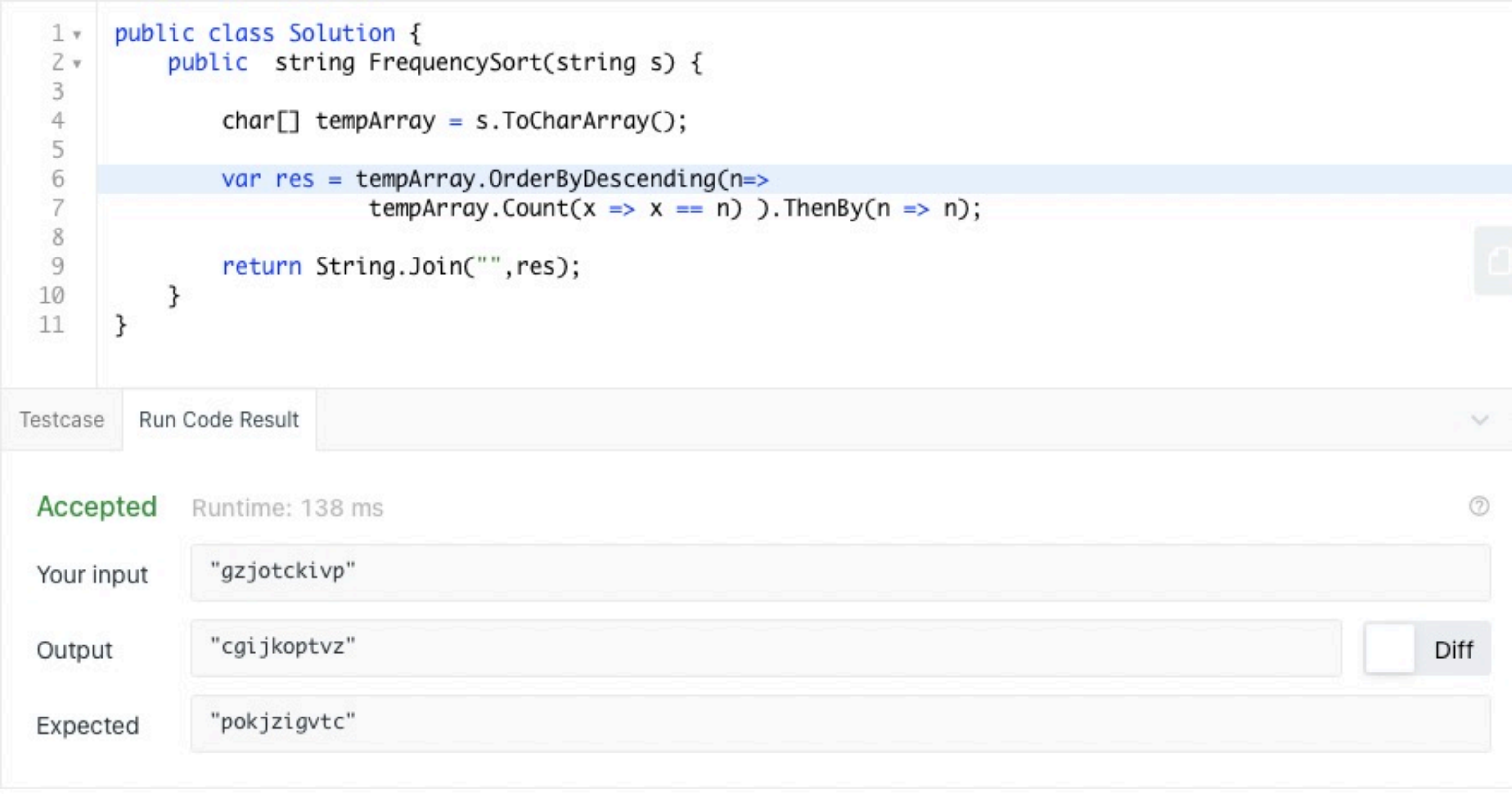
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 135 ms

Your input
```
"hnlnxiupgt"
```

Output
```
"nnghilptux"
```

Expected
```
"nnglixhtup"
```



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

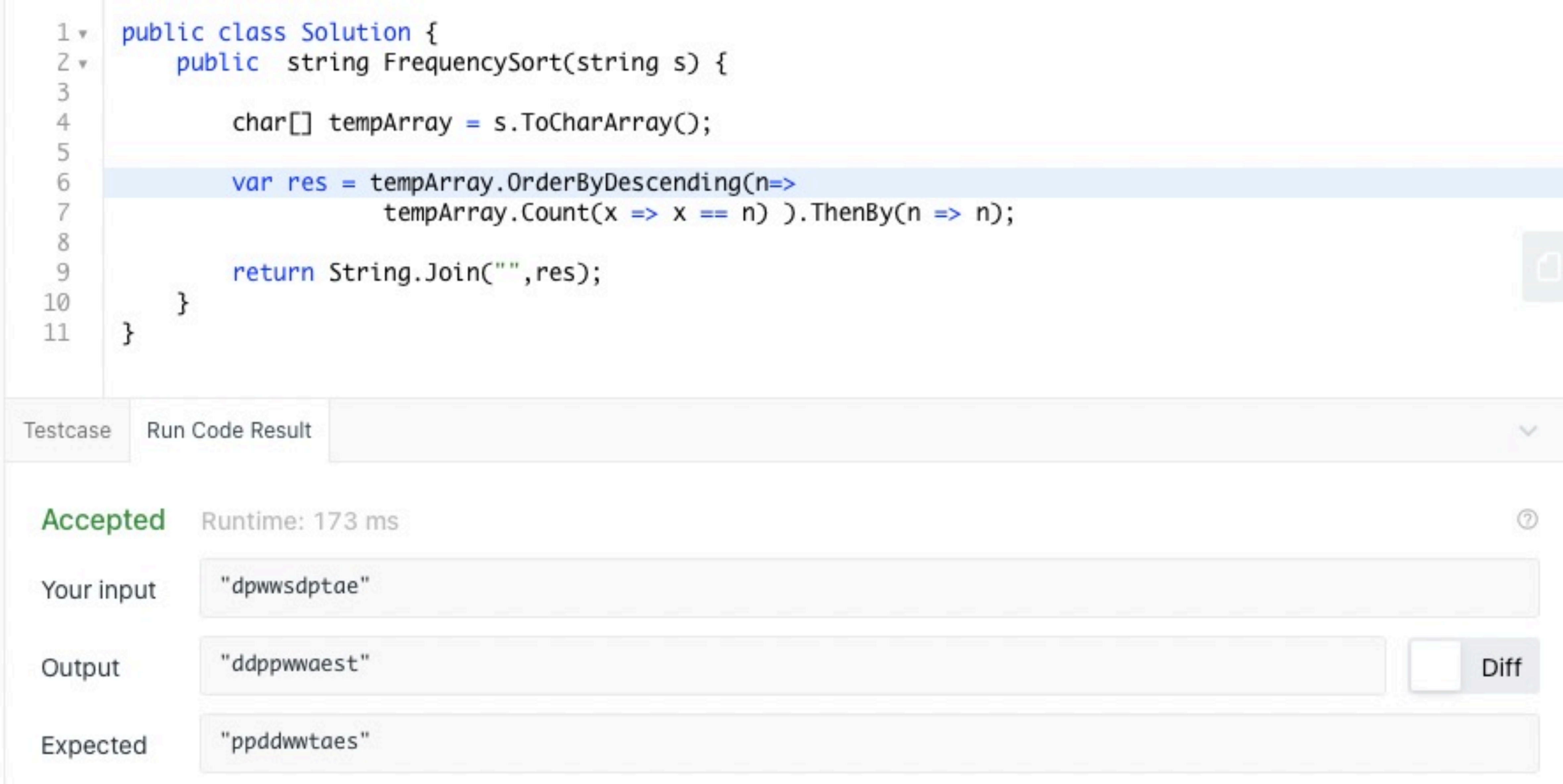
Testcase | Run Code Result

Accepted Runtime: 138 ms

Your input "gzjotckivp"

Output "cgijkoptvz" Diff

Expected "pokjzigvtc"



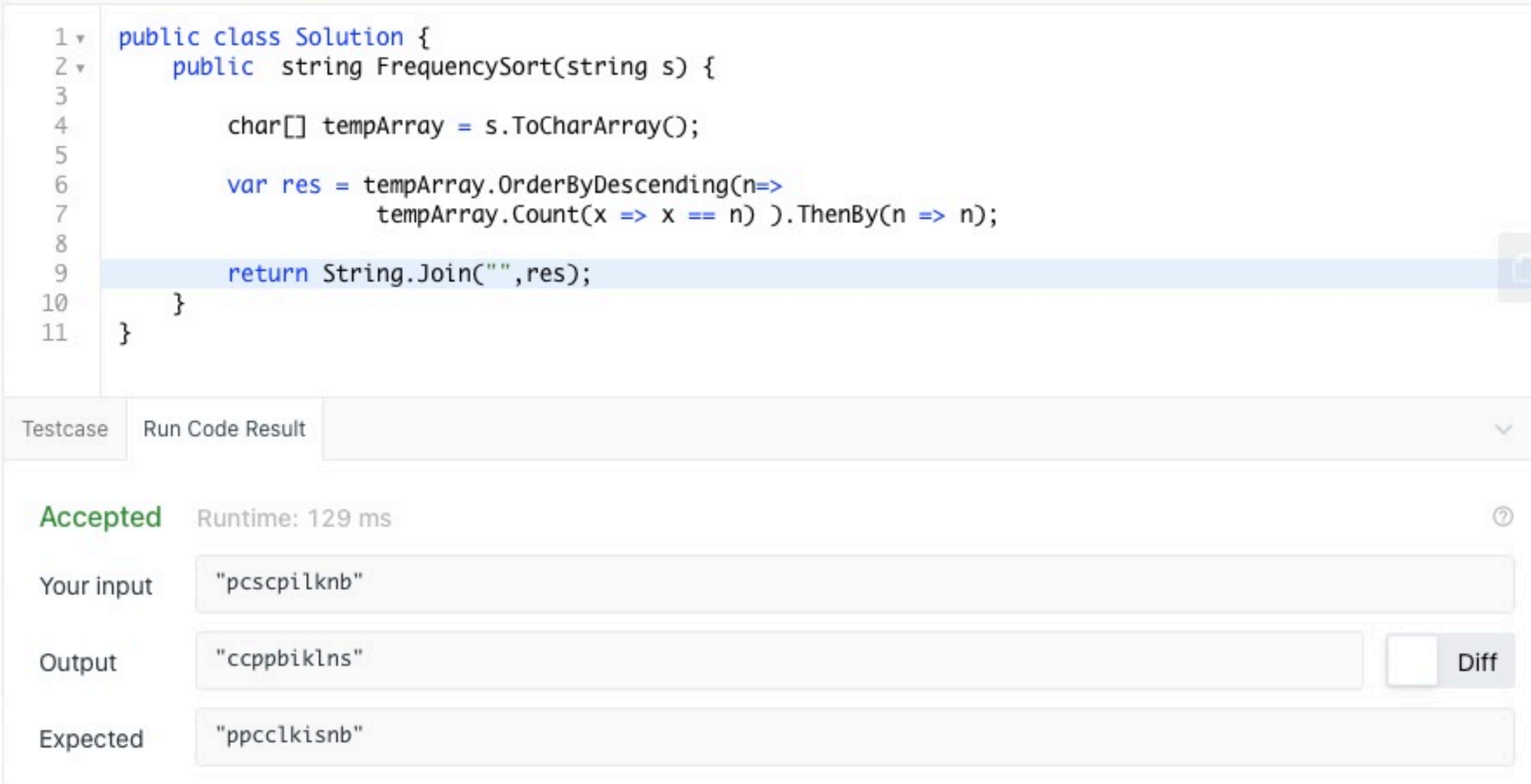
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 173 ms

Your input "dpwwsdptae"

Output "ddppwwaest"

Expected "ppddwwtaes"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Testcase | Run Code Result

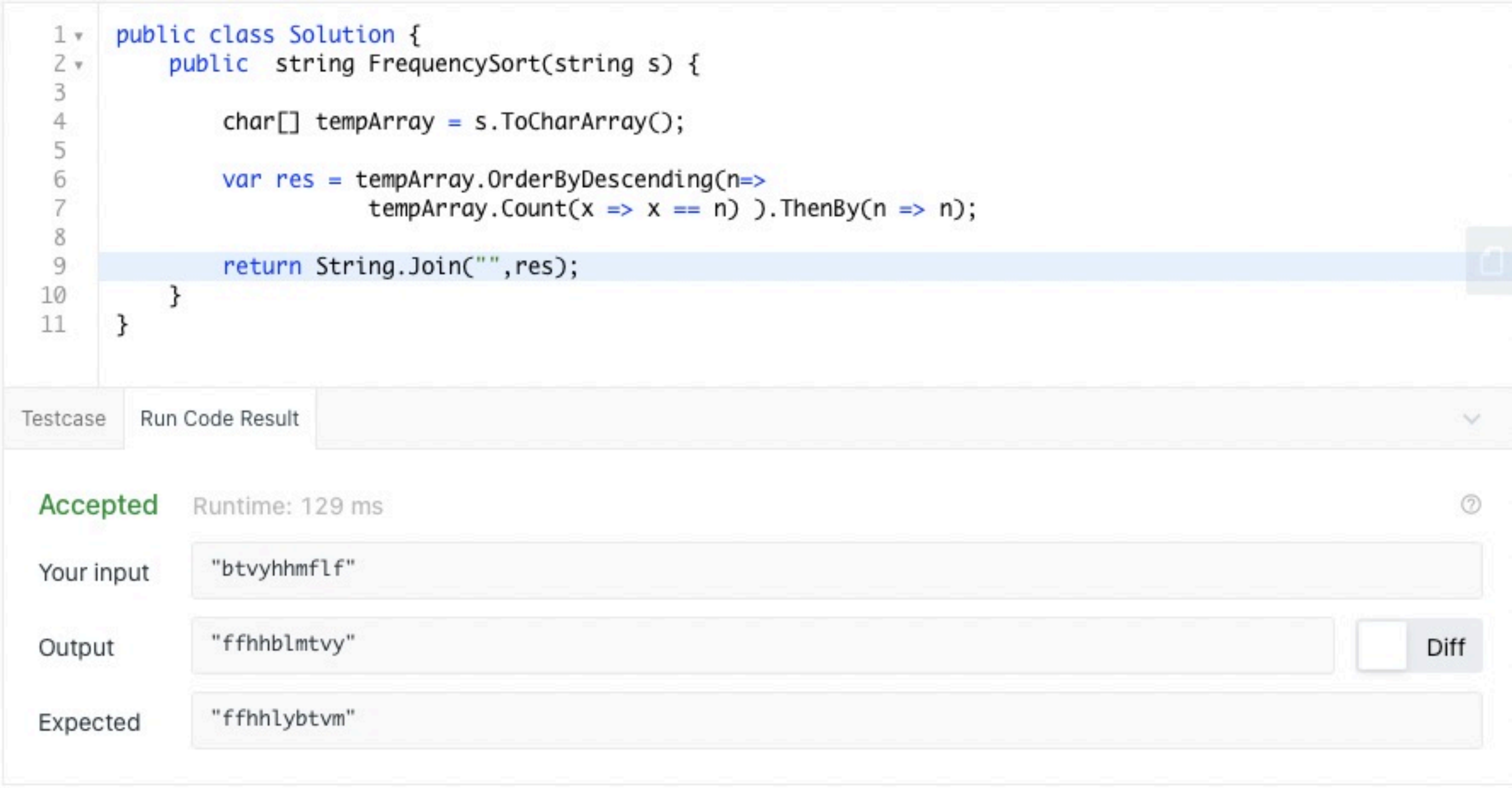
Accepted Runtime: 129 ms

Your input
```
"pcscpilknb"
```

Output
```
"ccppbiklns"
```

Diff

Expected
```
"ppcclkisnb"
```



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

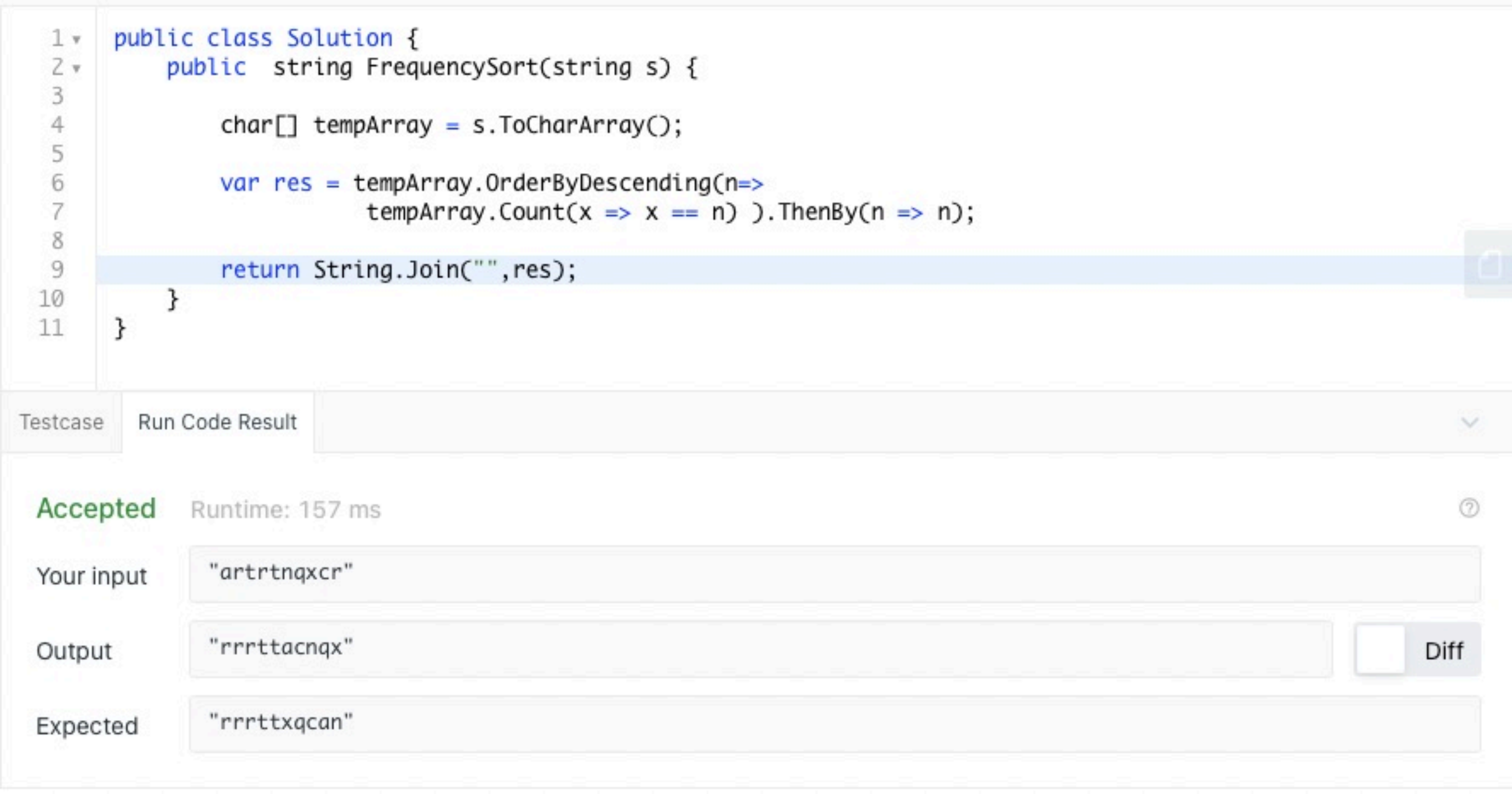
Testcase   Run Code Result

Accepted   Runtime: 129 ms

Your input   "btvyhhmflf"

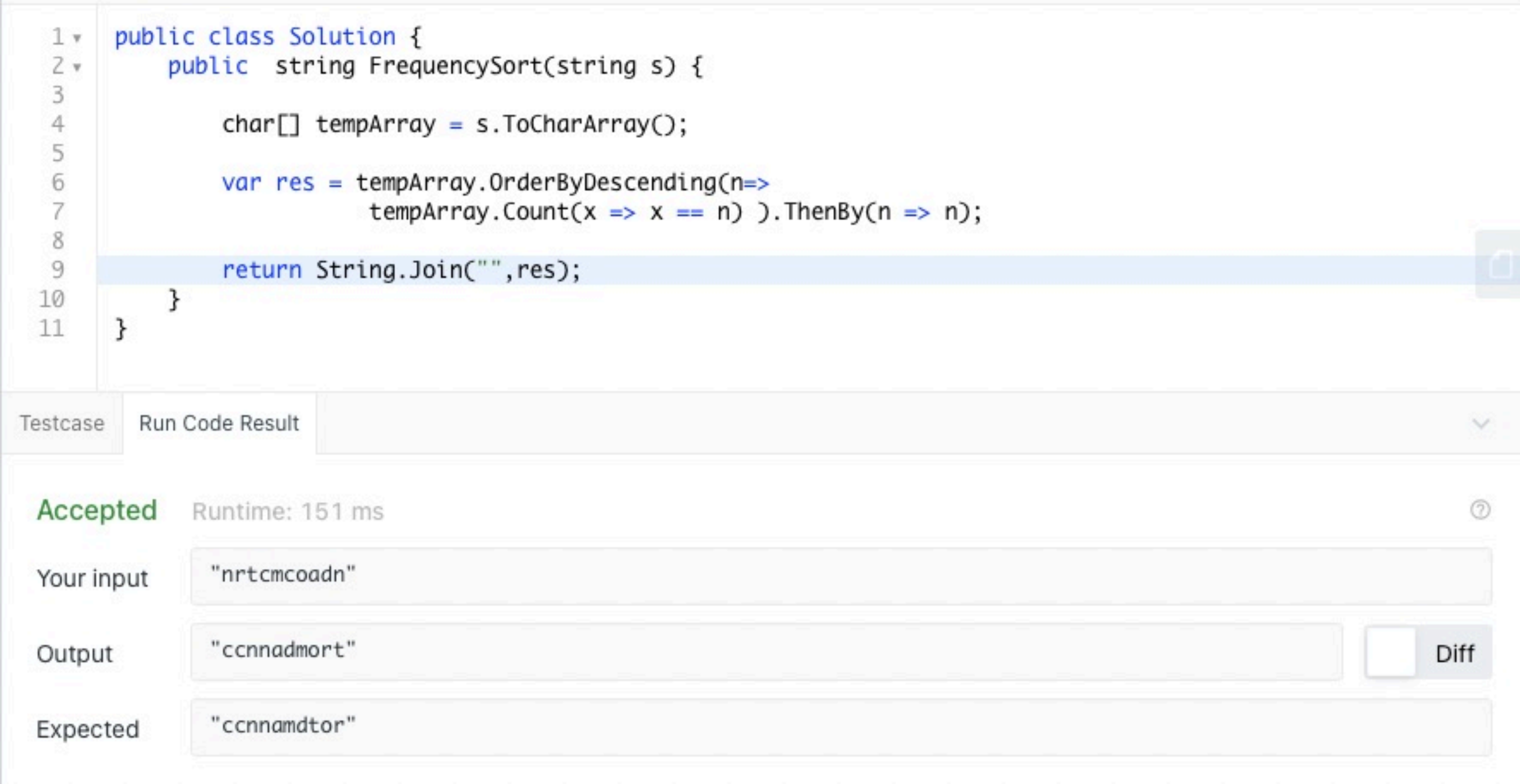
Output   "ffhhblmtvy"   Diff

Expected   "ffhhlybtvm"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

```
Accepted    Runtime: 157 ms

Your input  "artrtnqxcr"

Output      "rrrttacnqx"

Expected    "rrrttxqcan"
```



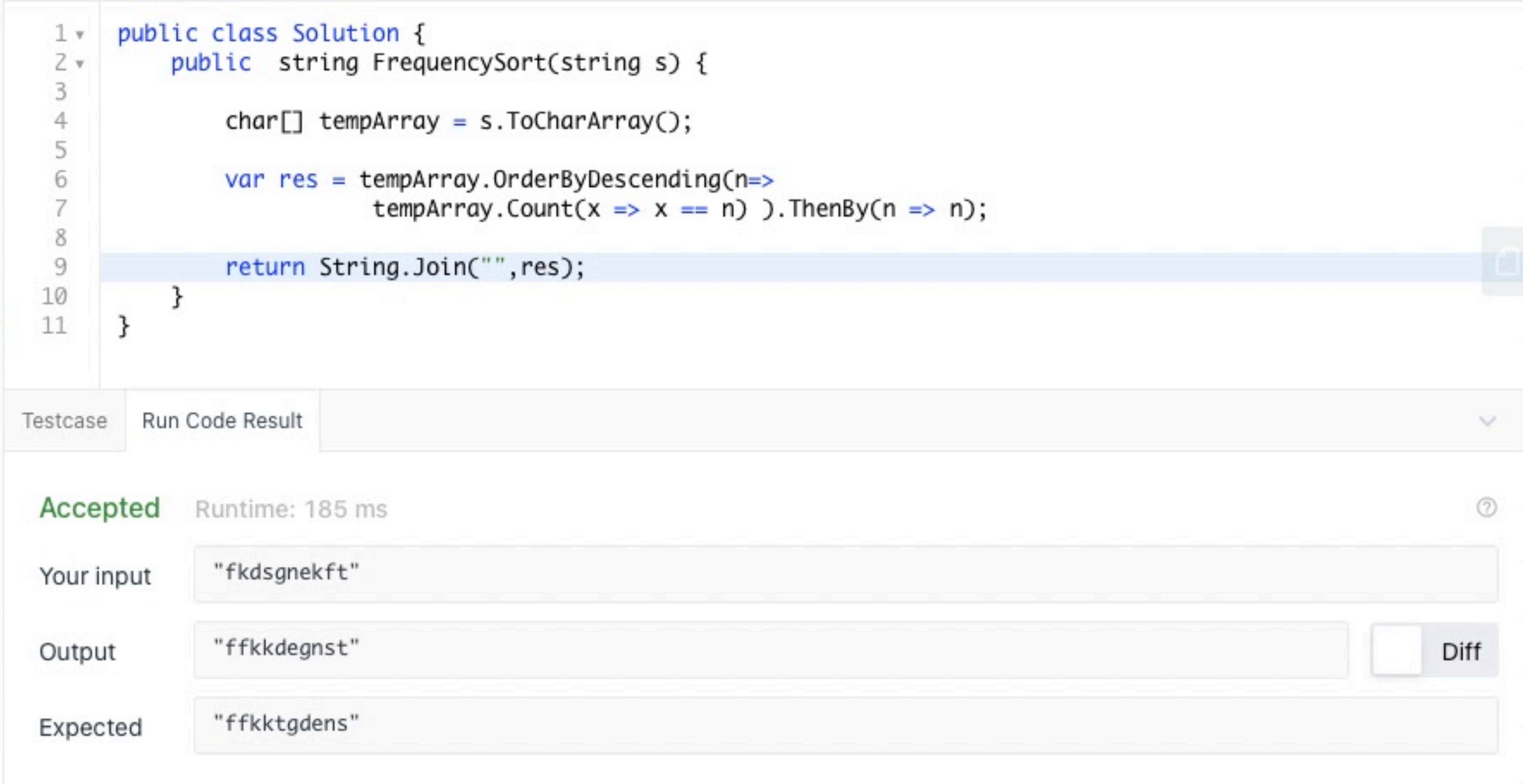
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 151 ms

Your input "nrtcmcoadn"

Output "ccnnadmort"

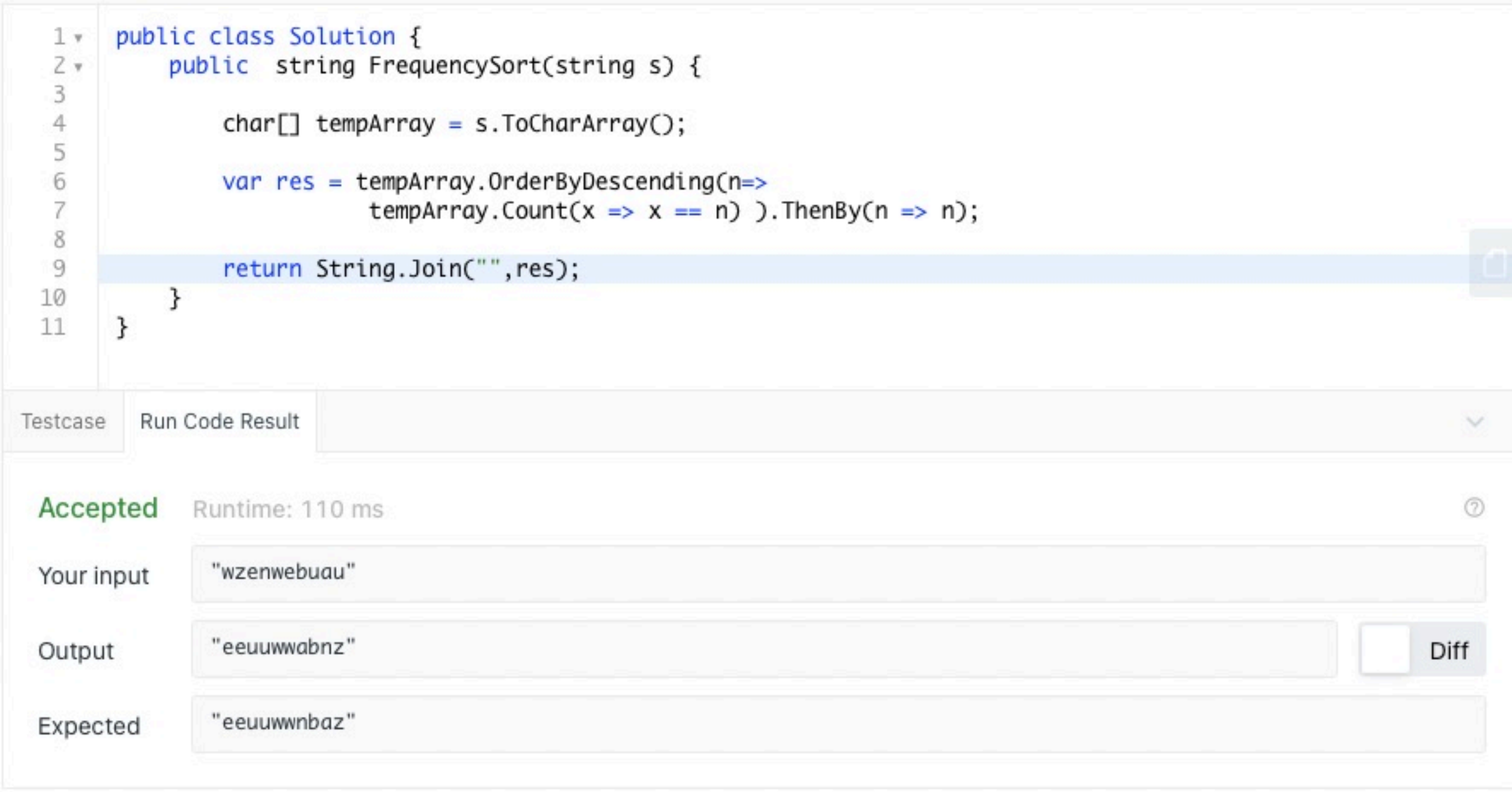
Expected "ccnnamdtor"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 185 ms

Your input	"fkdsgnekft"
Output	"ffkkdegnst"
Expected	"ffkktgdens"



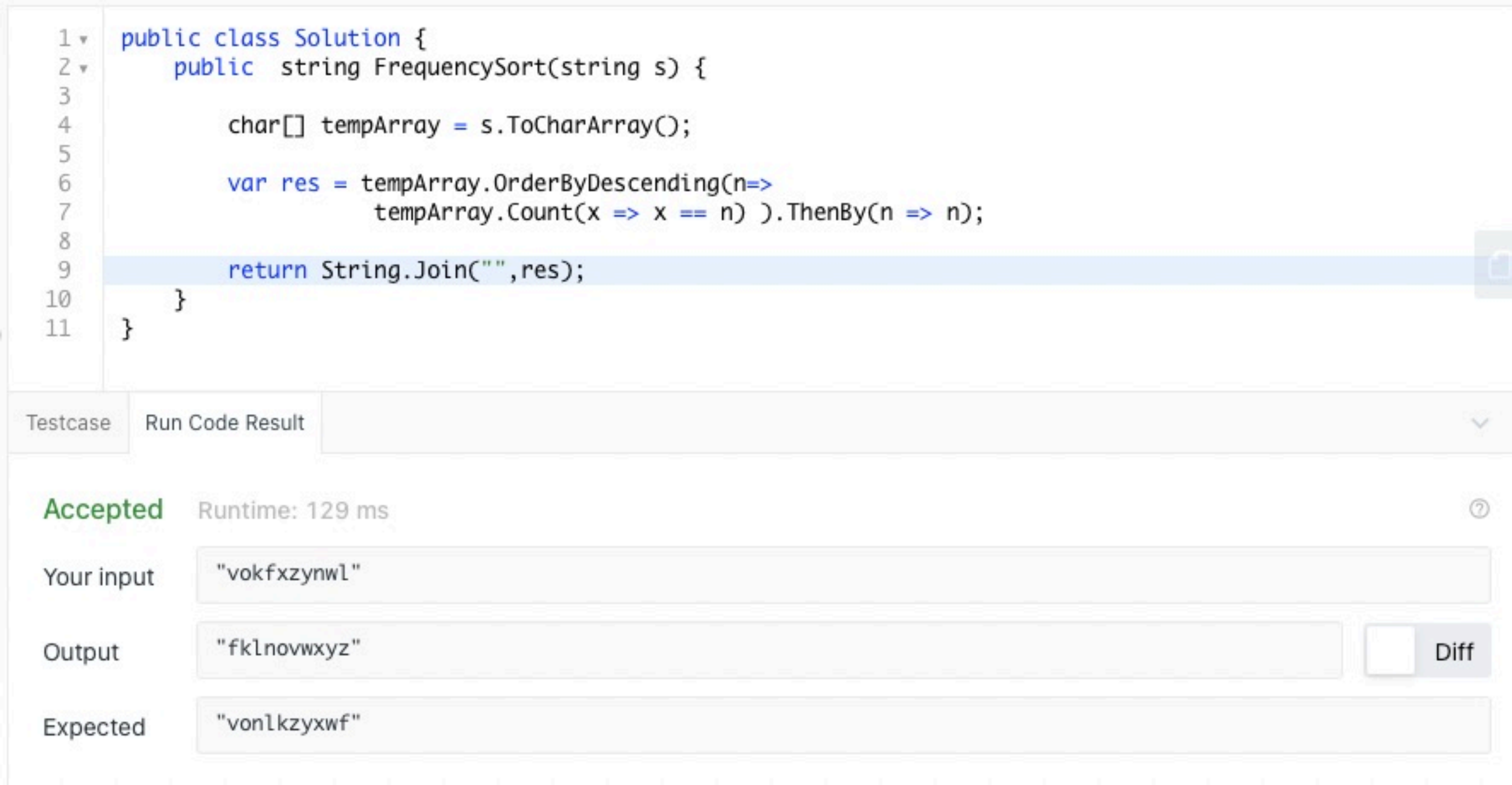
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 110 ms

Your input "wzenwebuau"

Output "eeuuwwabnz" Diff

Expected "eeuuwwnbaz"



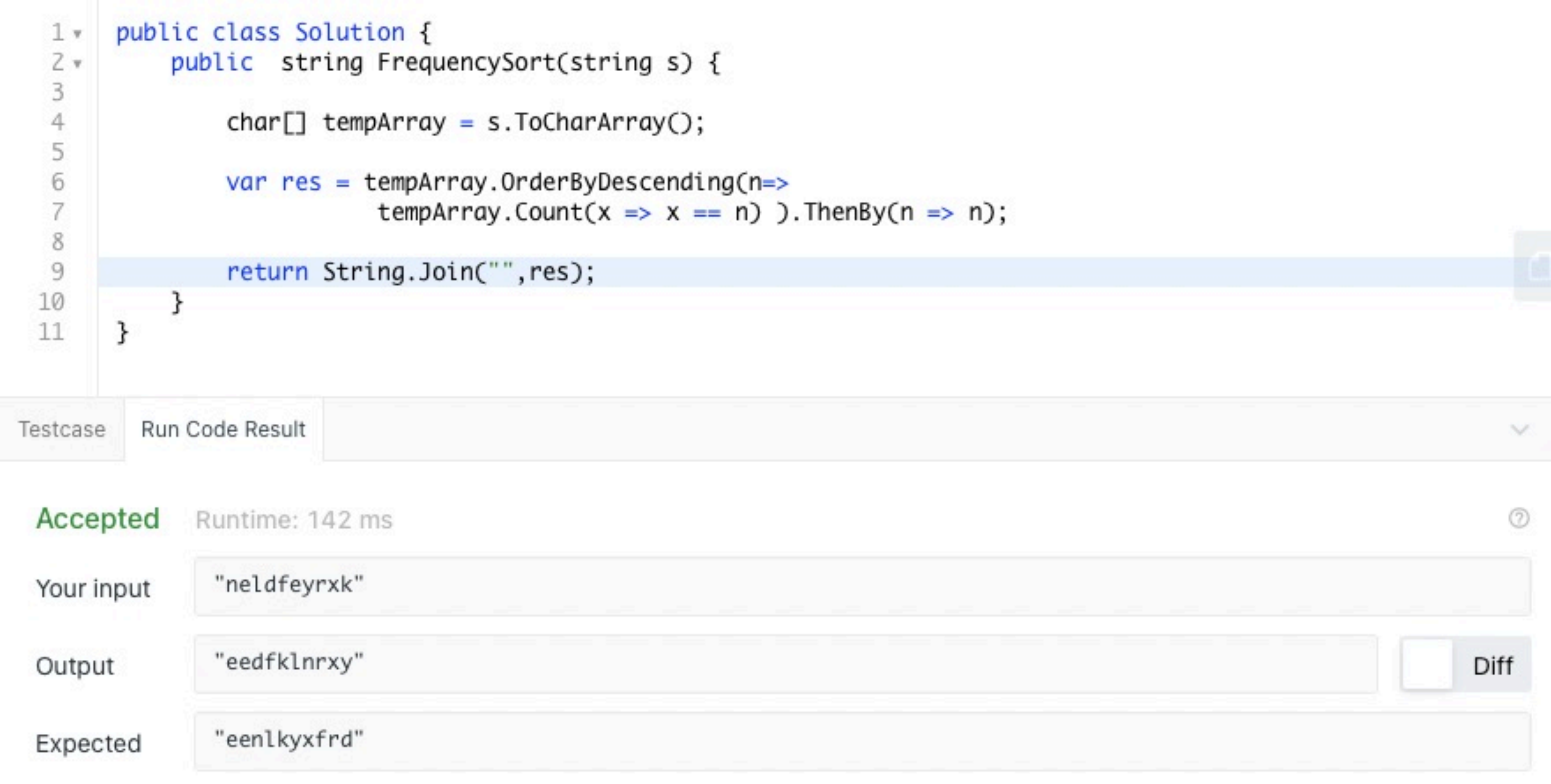
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 129 ms

Your input "vokfxzynwl"

Output "fklnovwxyz"

Expected "vonlkzyxwf"



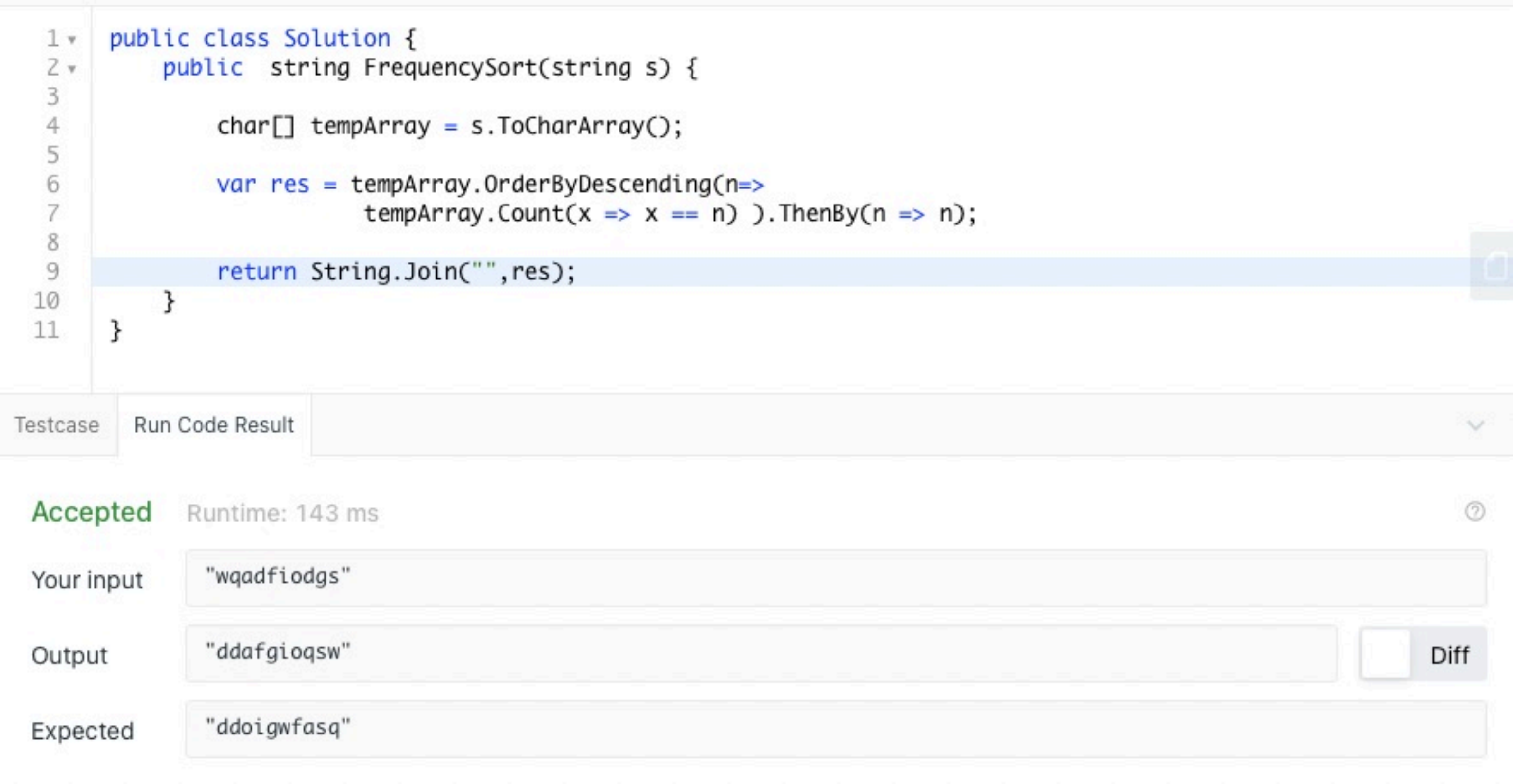
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 142 ms

Your input "neldfeyrxk"

Output "eedfklnrxy"

Expected "eenlkyxfrd"



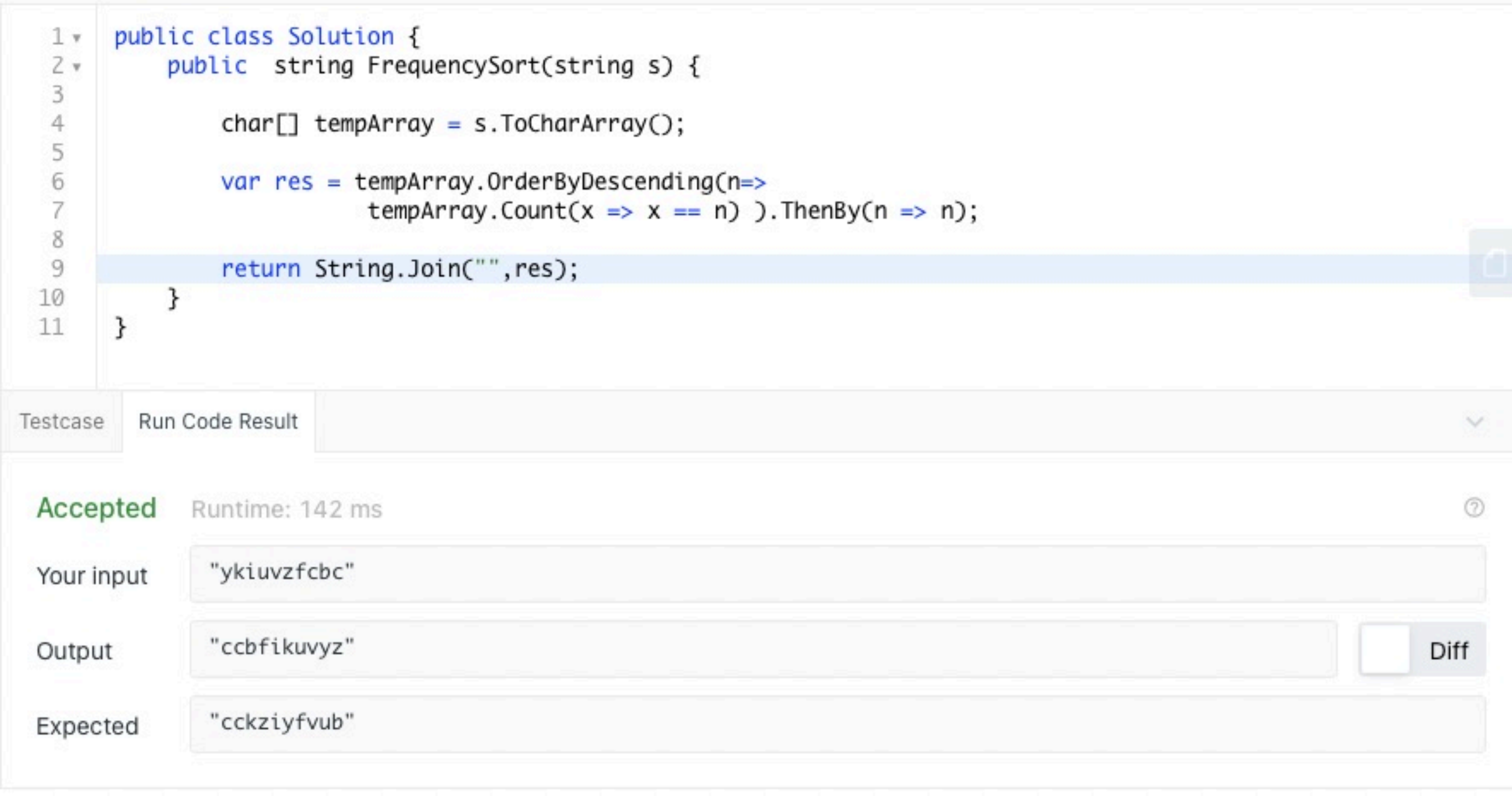
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 143 ms

Your input "wqadfiodgs"

Output "ddafgioqsw"

Expected "ddoigwfasq"



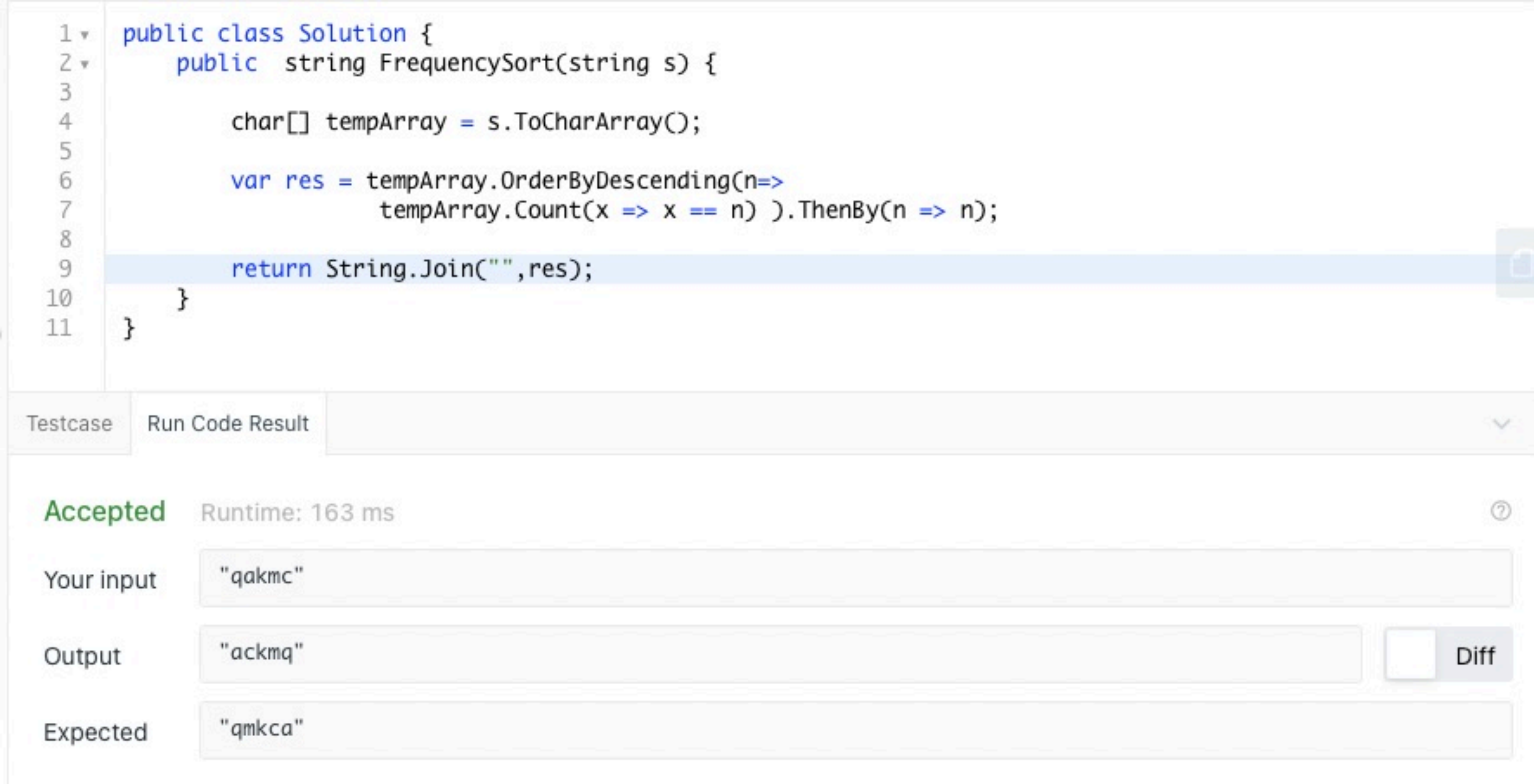
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 142 ms

Your input "ykiuvzfcbc"

Output "ccbfikuvyz"

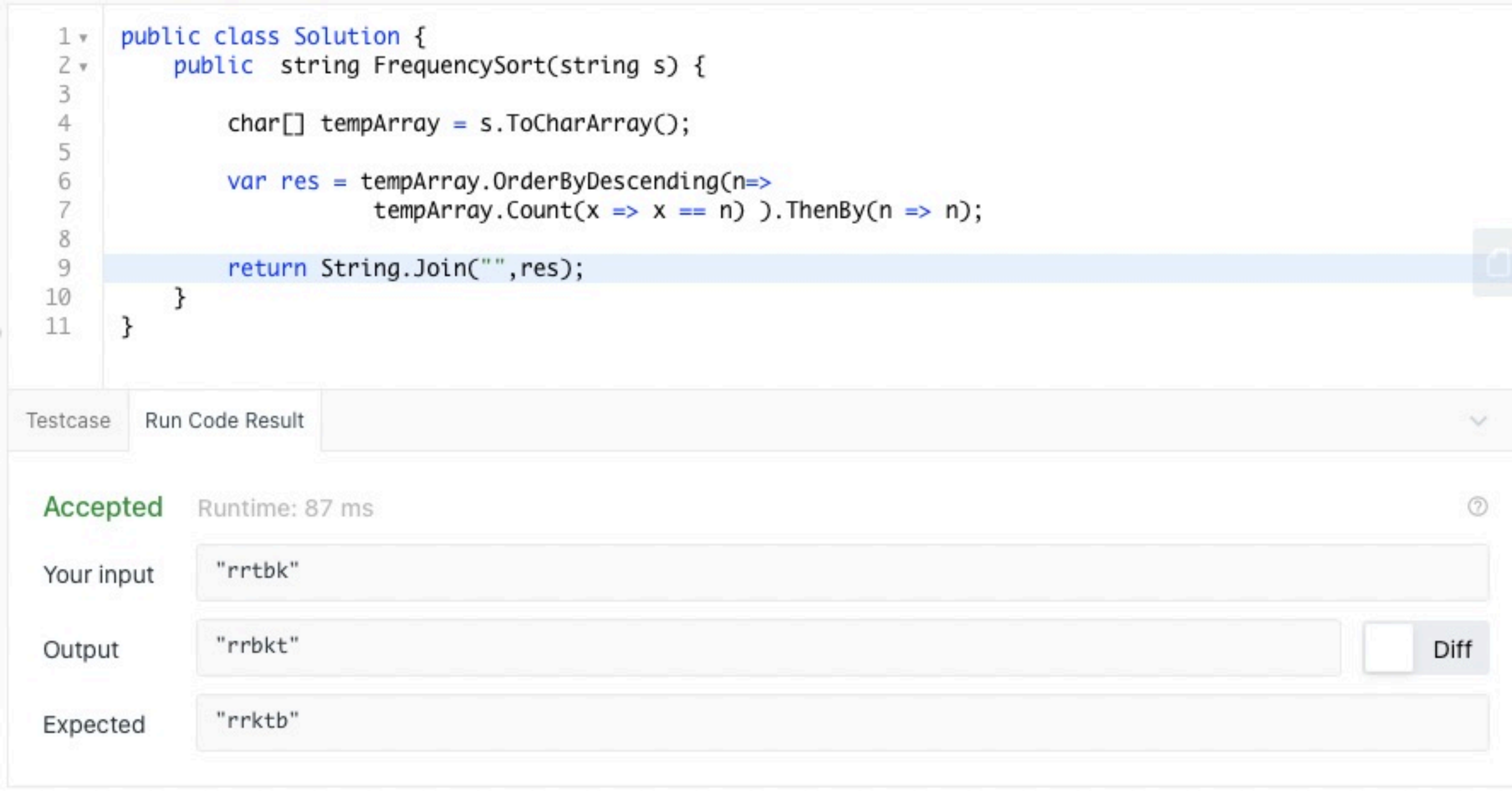
Expected "cckziyfvub"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 163 ms

Your input	"qakmc"
Output	"ackmq"
Expected	"qmkca"



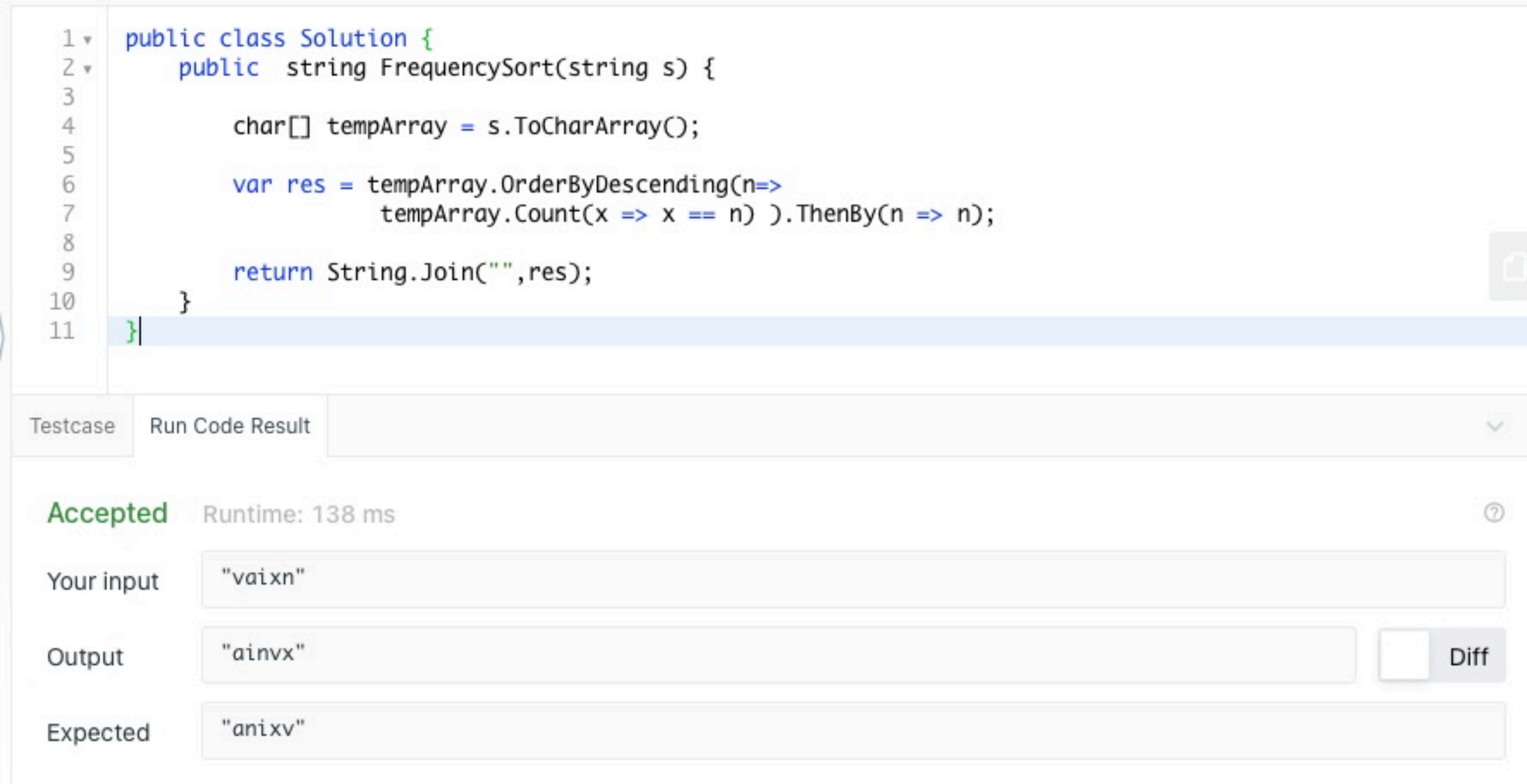
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 87 ms

Your input "rrtbk"

Output "rrbkt"

Expected "rrktb"



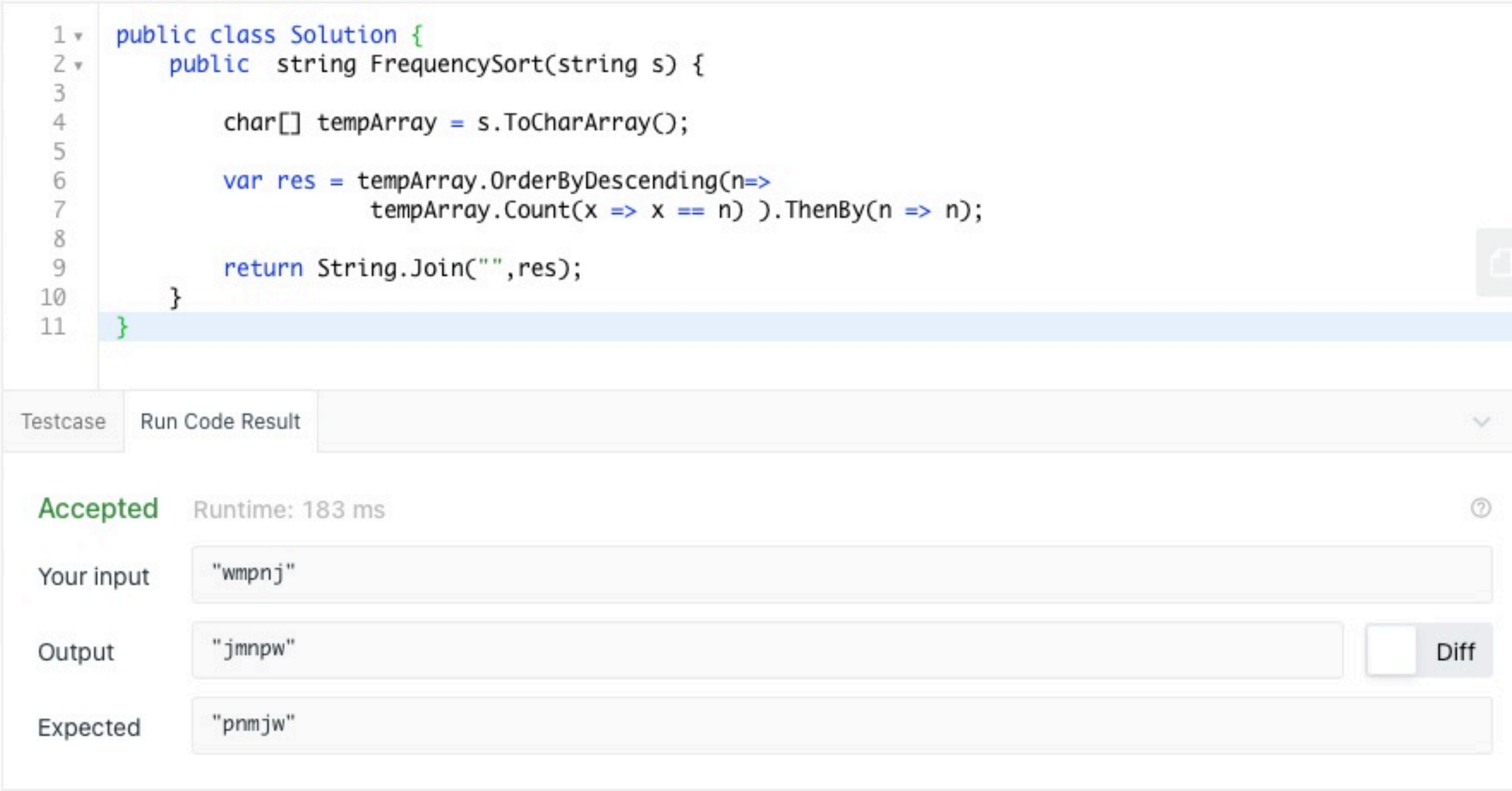
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 138 ms

Your input "vaixn"

Output "ainvx" Diff

Expected "anixv"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                      tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

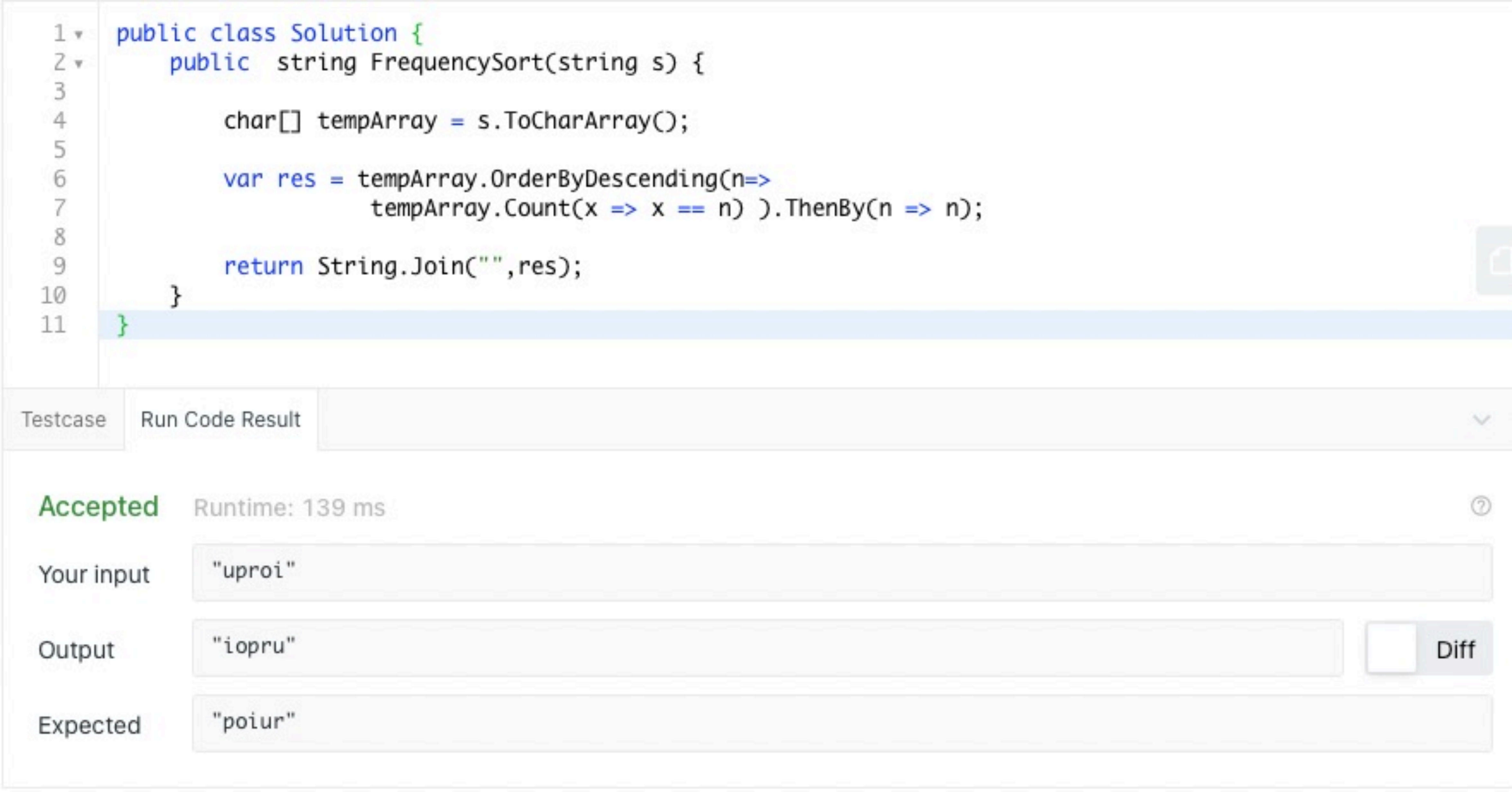
Testcase | Run Code Result

Accepted Runtime: 183 ms

Your input "wmpnj"

Output "jmnpw" Diff

Expected "pnmjw"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

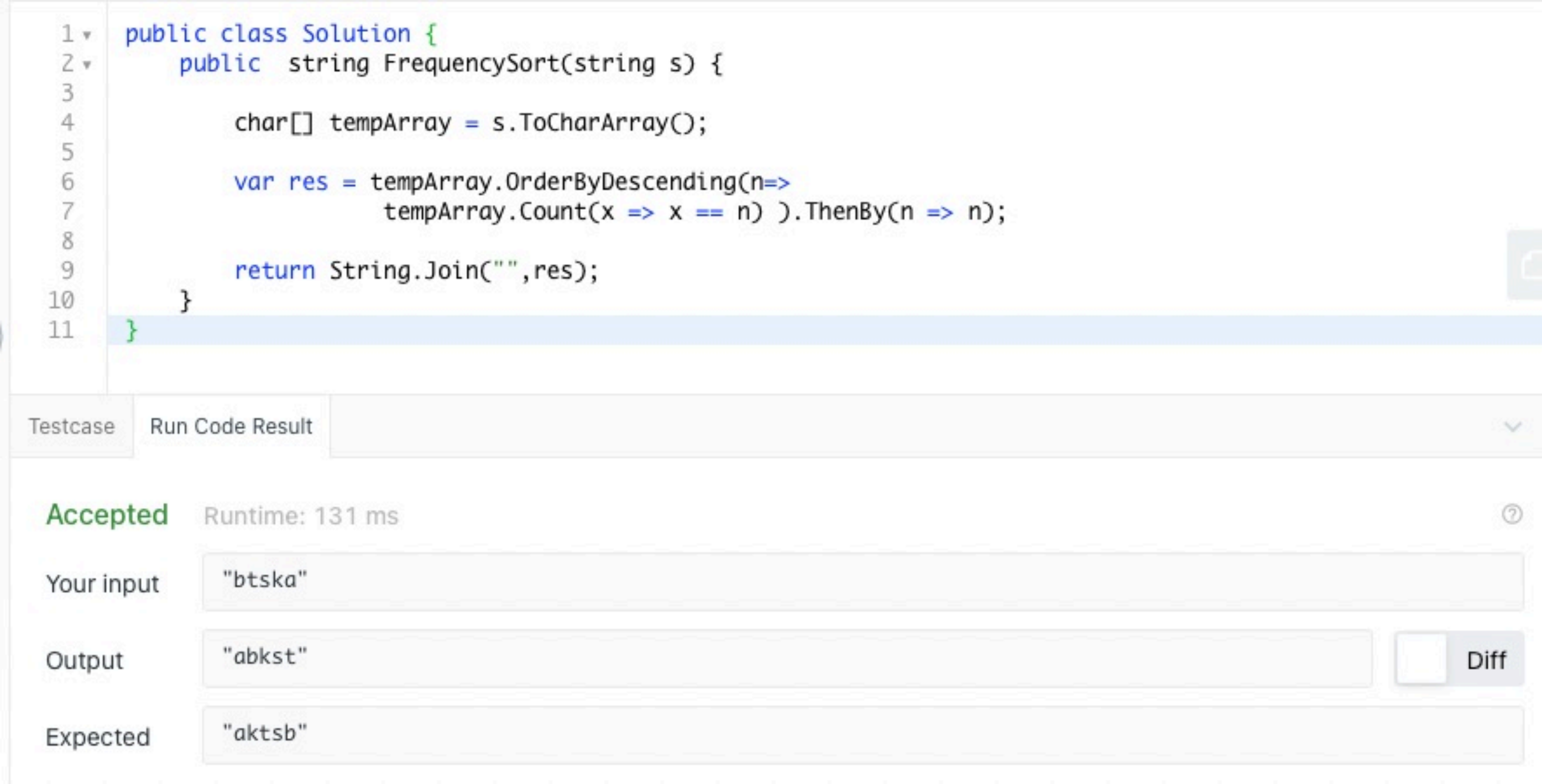
Testcase   Run Code Result

Accepted   Runtime: 139 ms

Your input   "uproi"

Output   "iopru"   Diff

Expected   "poiur"



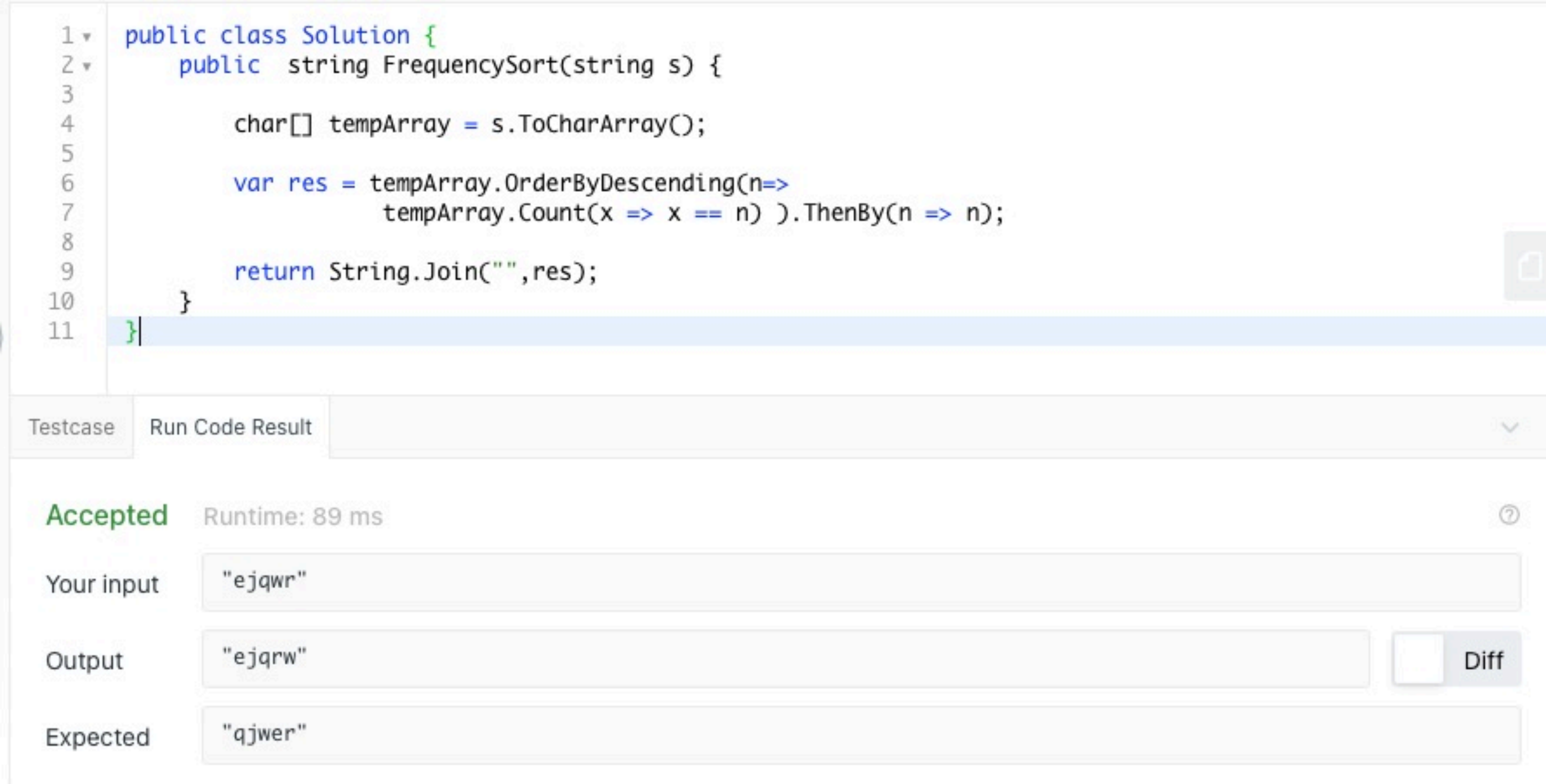
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 131 ms

Your input "btska"

Output "abkst" Diff

Expected "aktsb"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

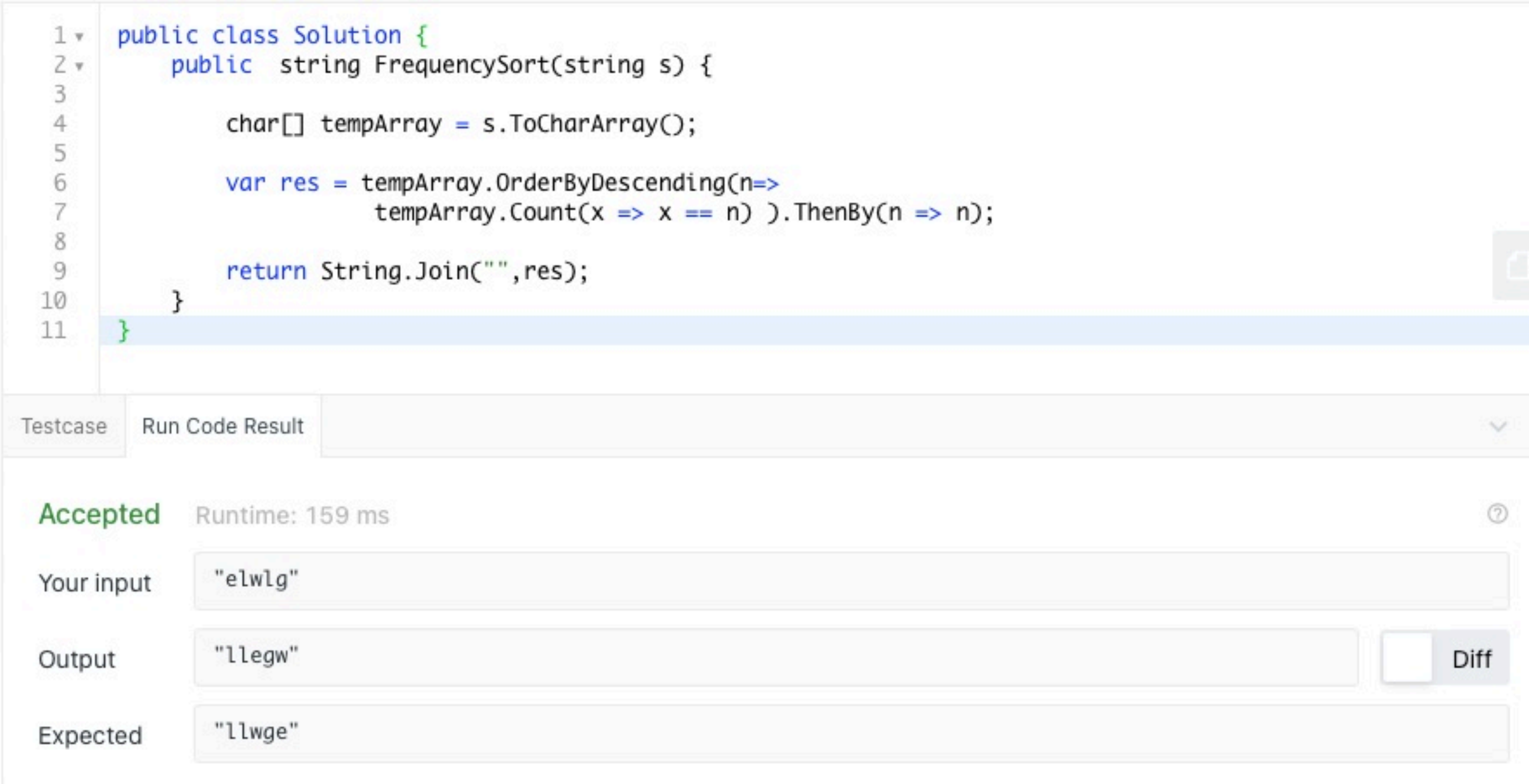
Testcase Run Code Result

Accepted Runtime: 89 ms

Your input "ejqwr"

Output "ejqrw" Diff

Expected "qjwer"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

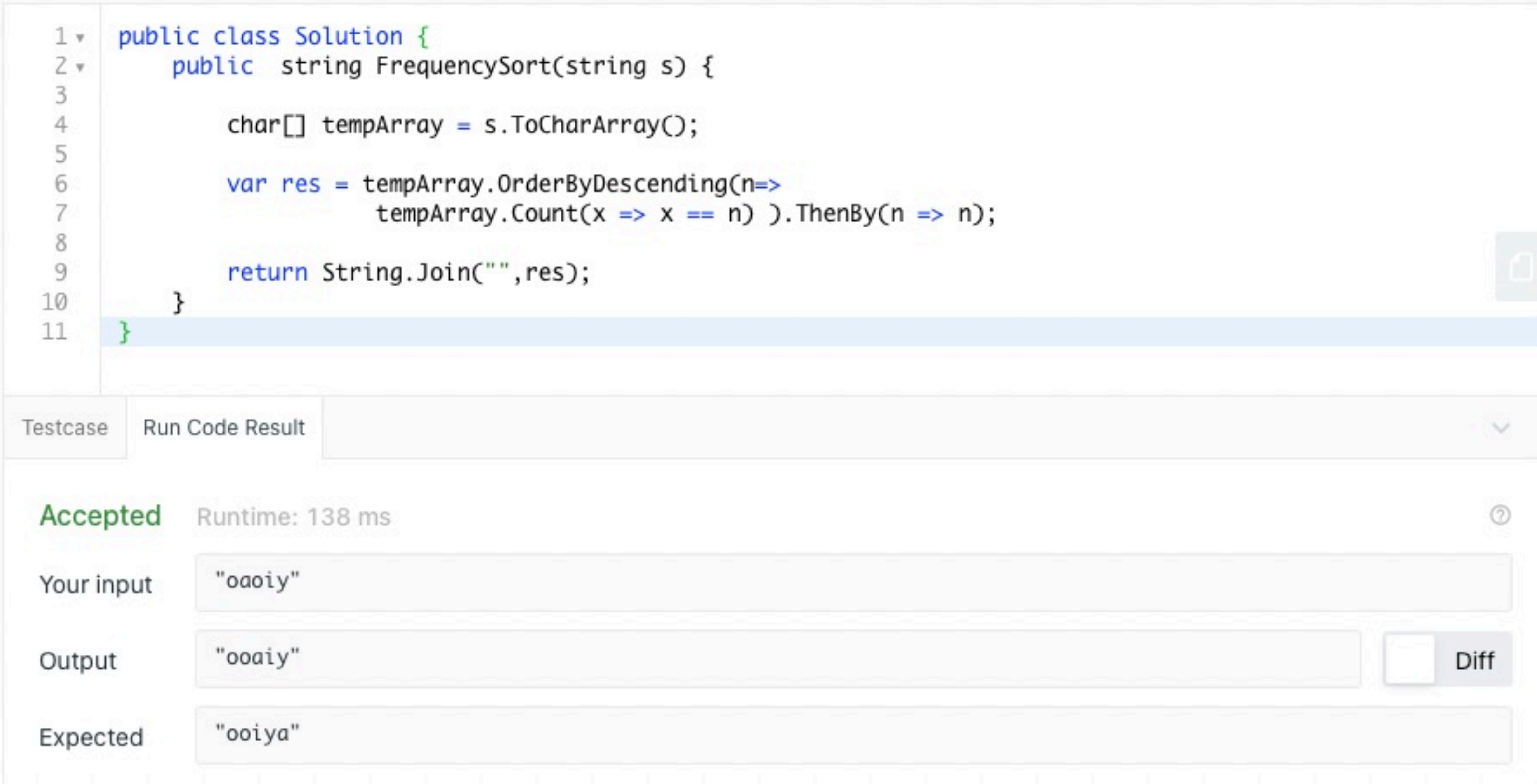
Testcase | Run Code Result

Accepted Runtime: 159 ms

Your input "elwlg"

Output "llegw" Diff

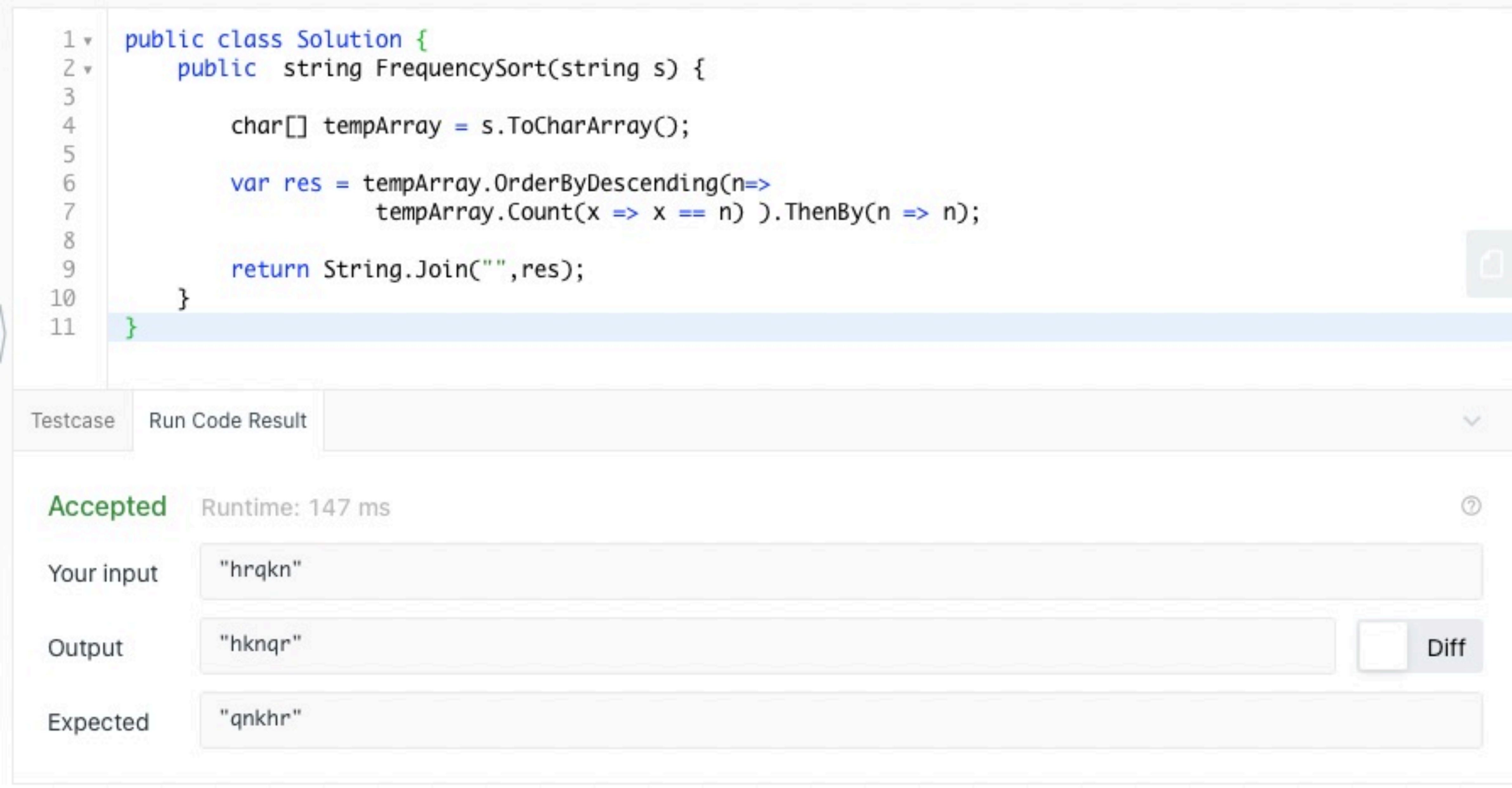
Expected "llwge"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 138 ms

Your input	"oaoiy"
Output	"ooaiy"
Expected	"ooiya"



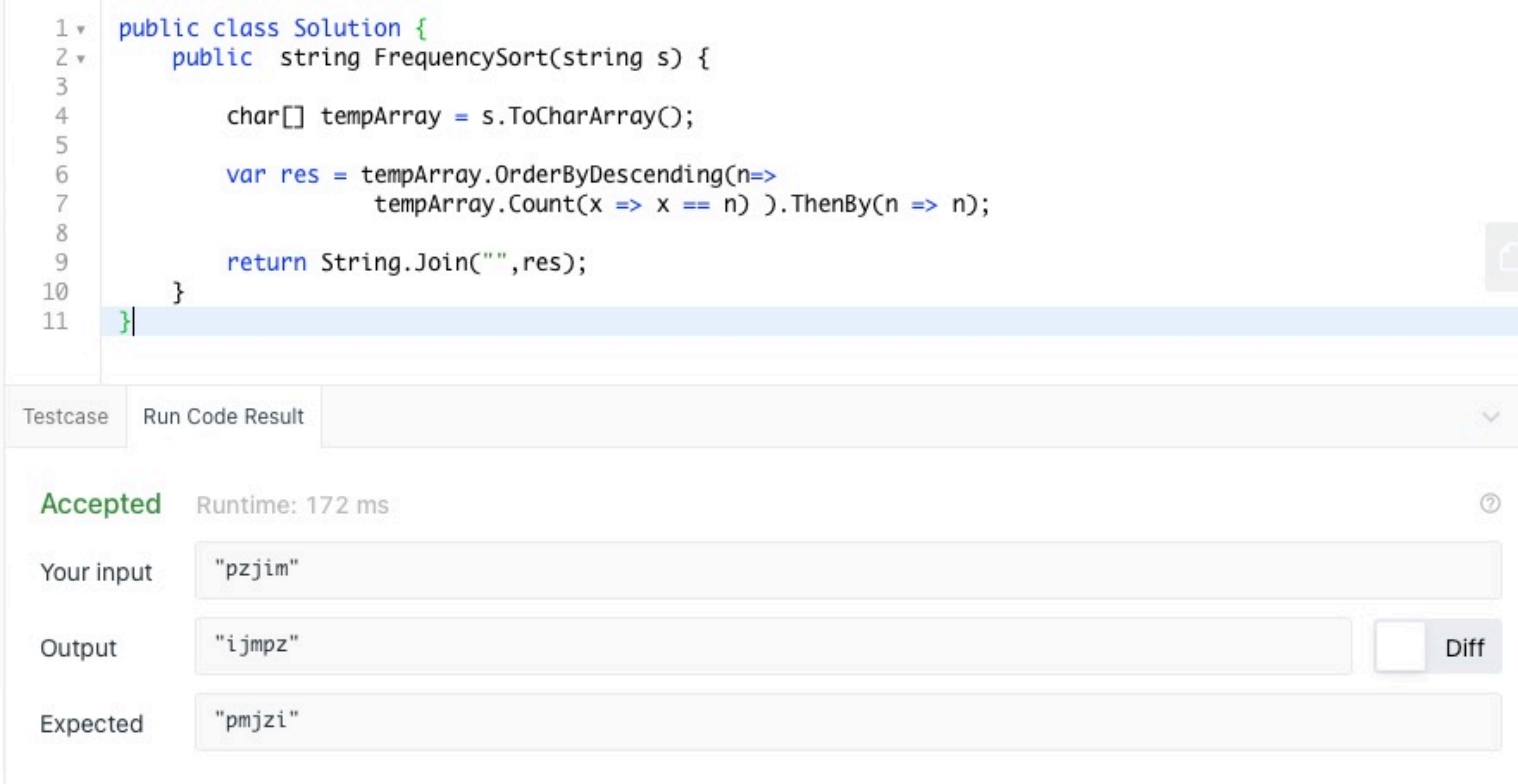
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 147 ms

Your input "hrqkn"

Output "hknqr"

Expected "qnkhr"



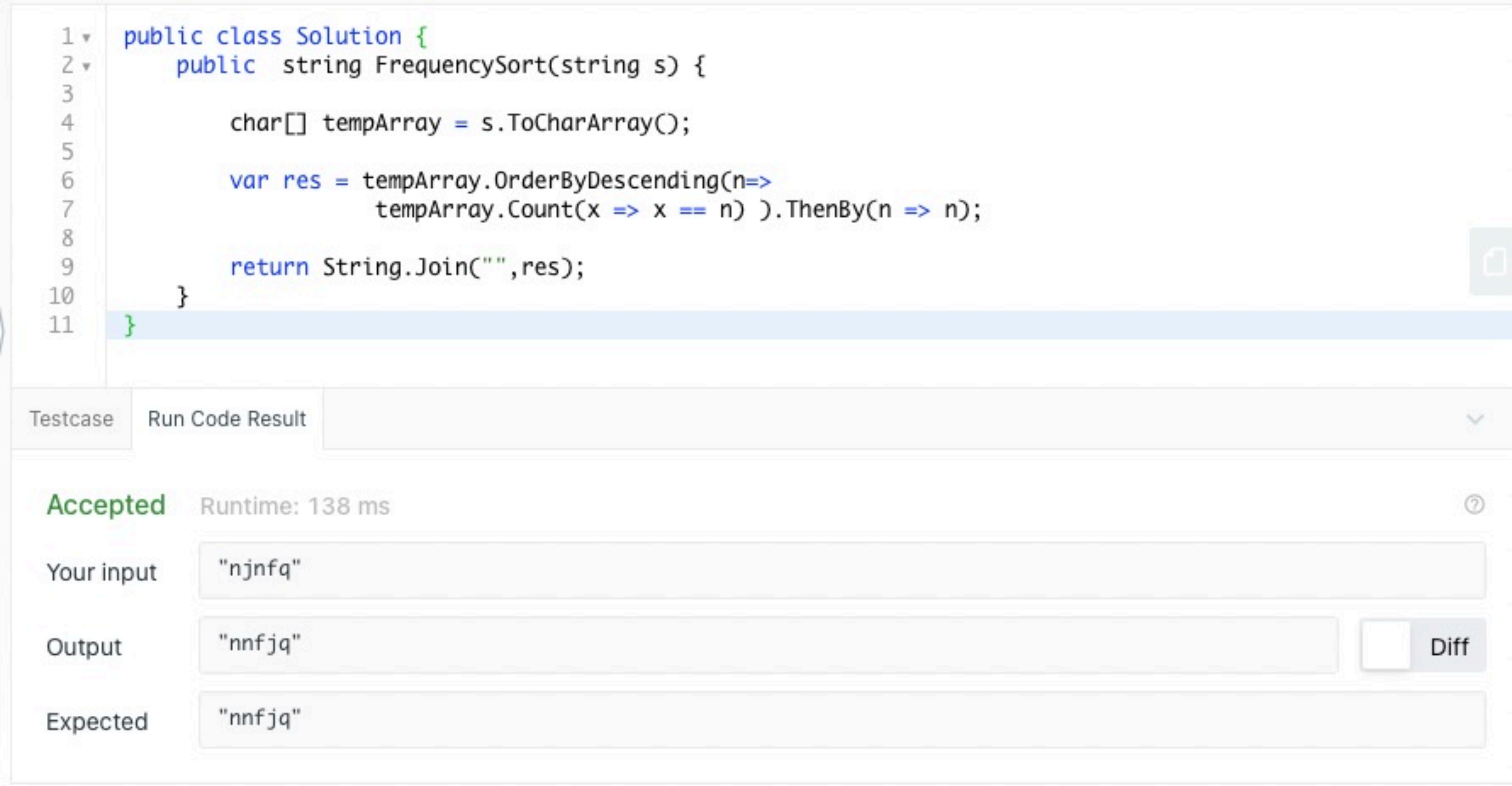
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 172 ms

Your input "pzjim"

Output "ijmpz"

Expected "pmjzi"



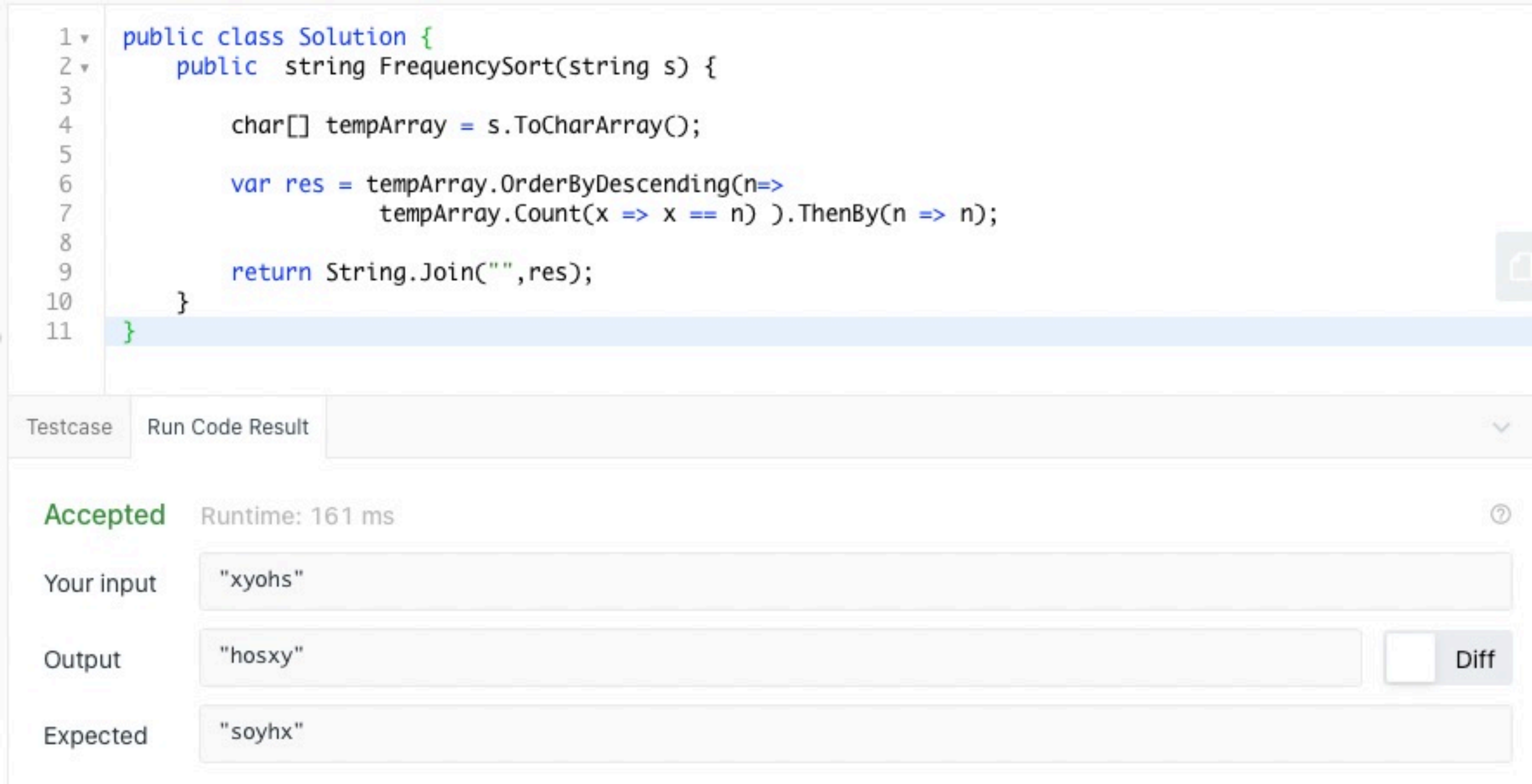
```csharp
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 138 ms

Your input "njnfq"

Output "nnfjq"

Expected "nnfjq"



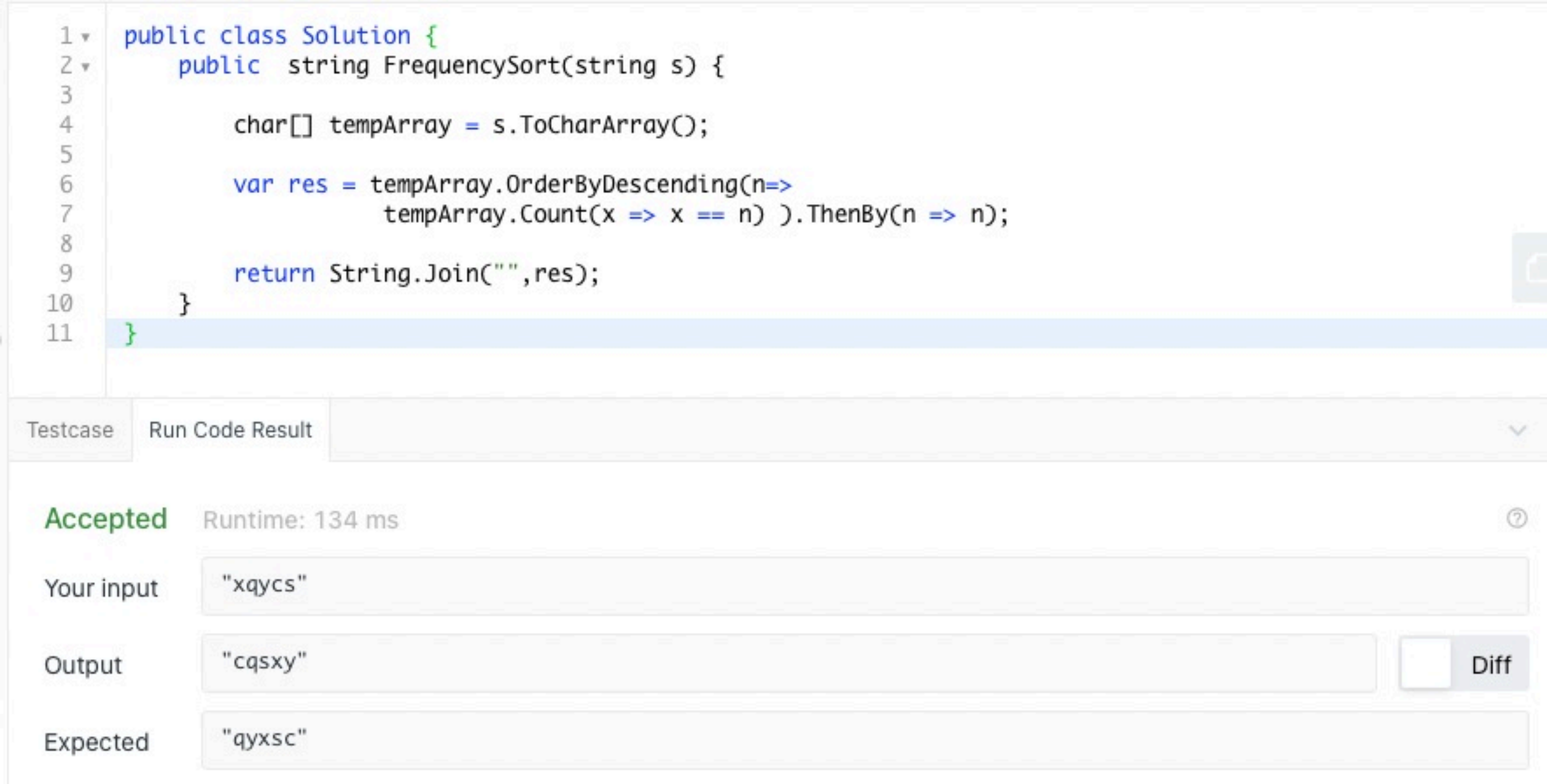
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 161 ms

Your input "xyohs"

Output "hosxy"

Expected "soyhx"


```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Testcase | Run Code Result

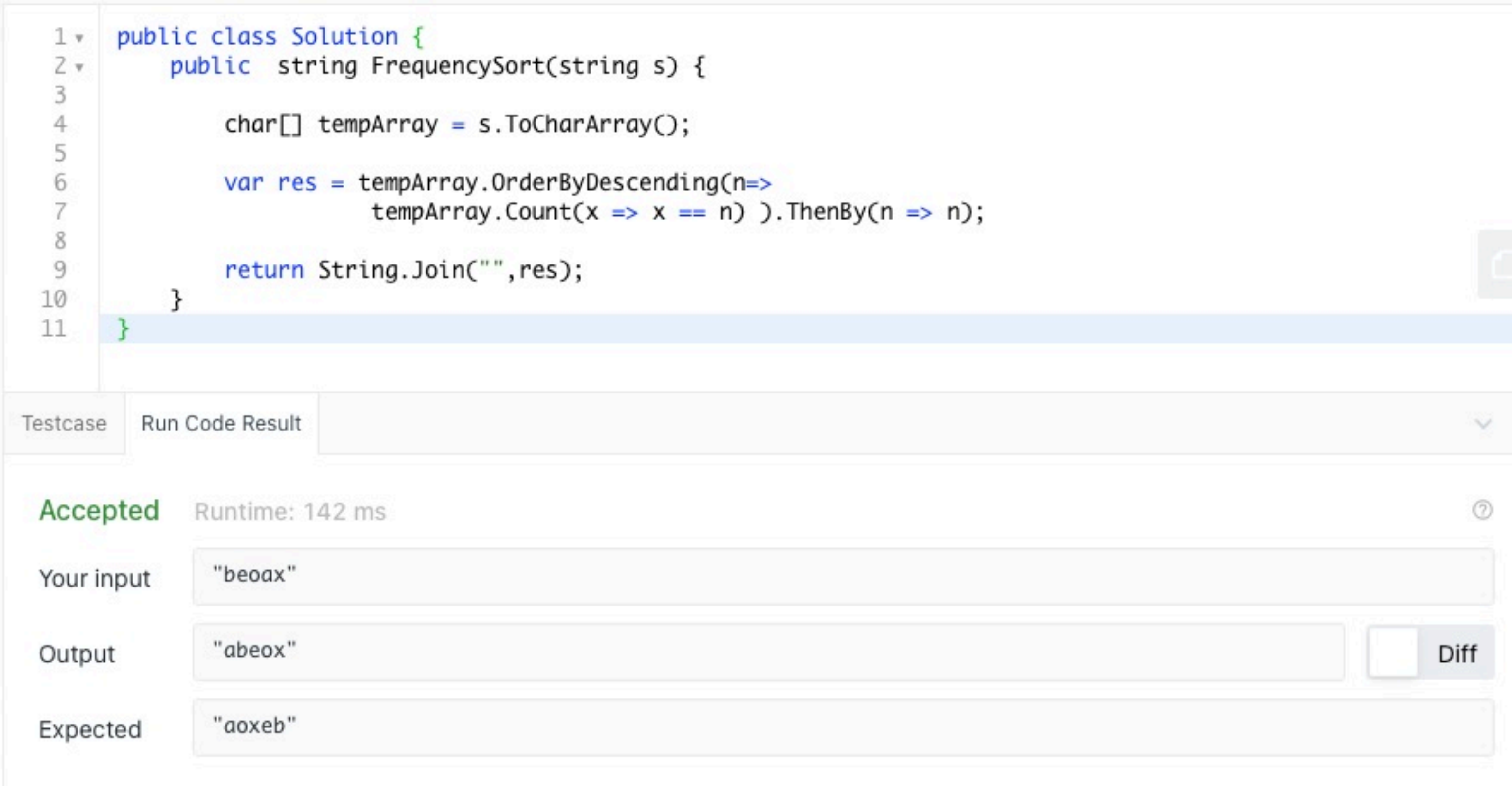
Accepted Runtime: 134 ms

Your input
```
"xqycs"
```

Output
```
"cqsxy"
```

Diff

Expected
```
"qyxsc"
```



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

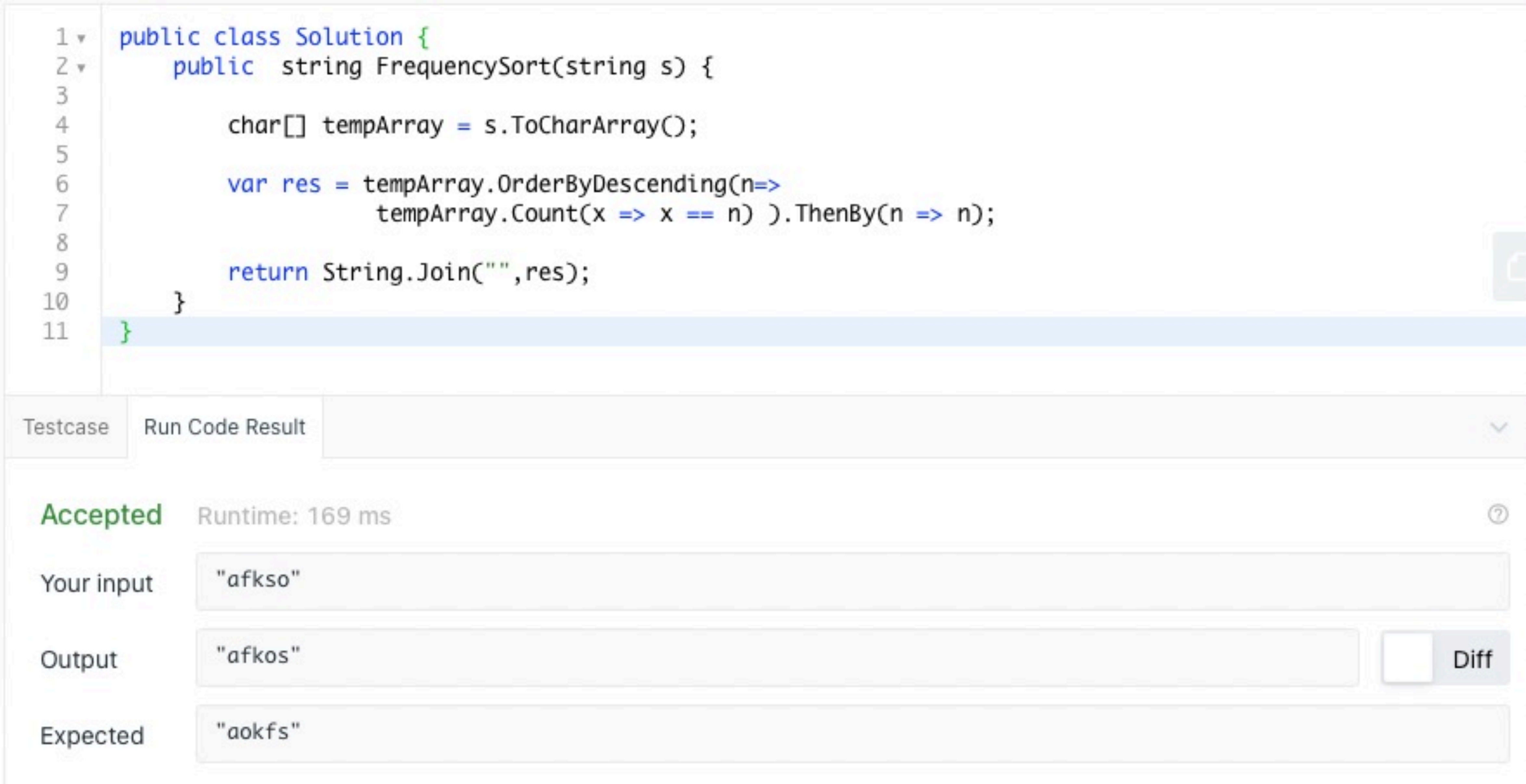
Testcase | Run Code Result

Accepted Runtime: 142 ms

Your input "beoax"

Output "abeox"

Expected "aoxeb"



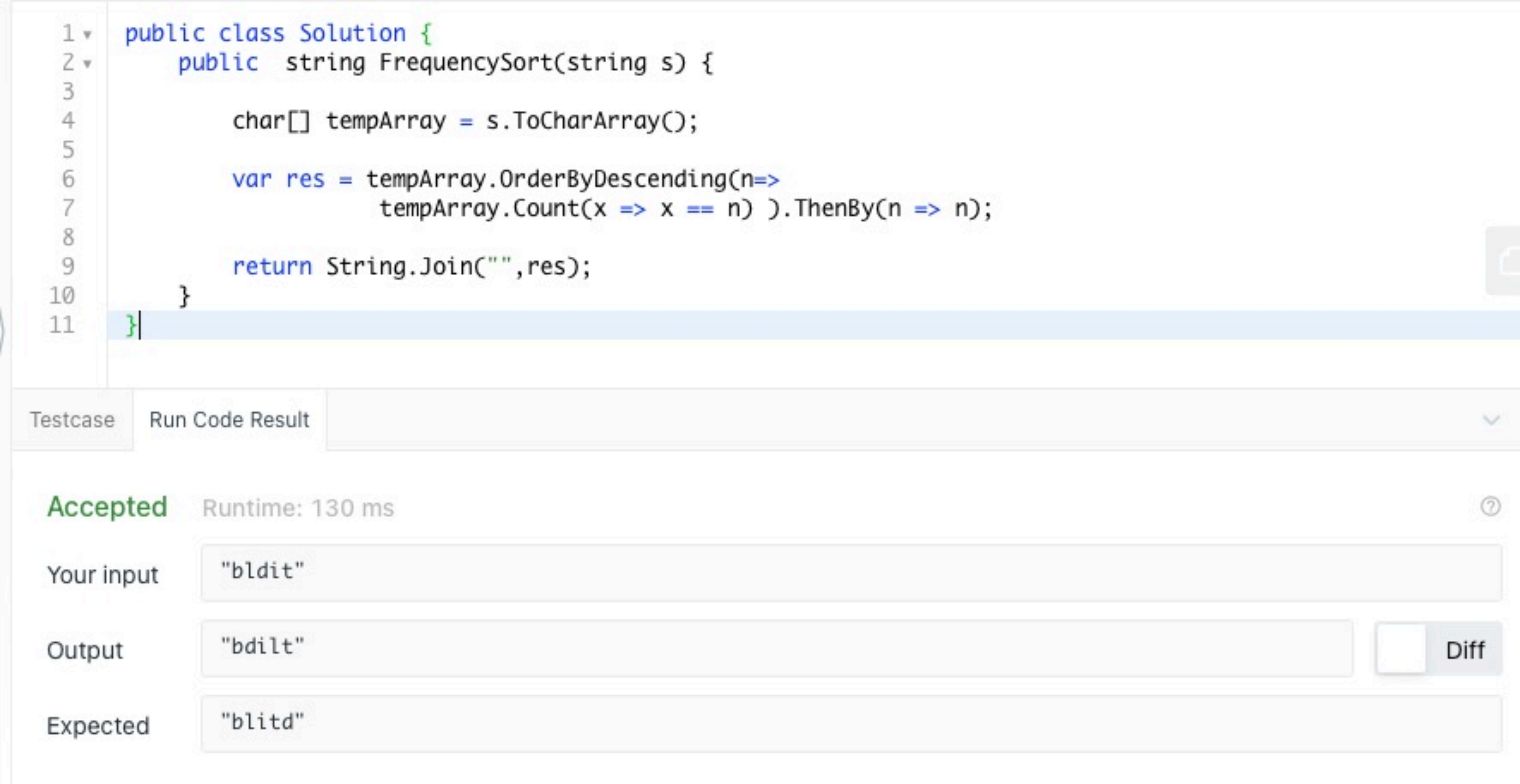
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                     tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 169 ms

Your input "afkso"

Output "afkos"

Expected "aokfs"



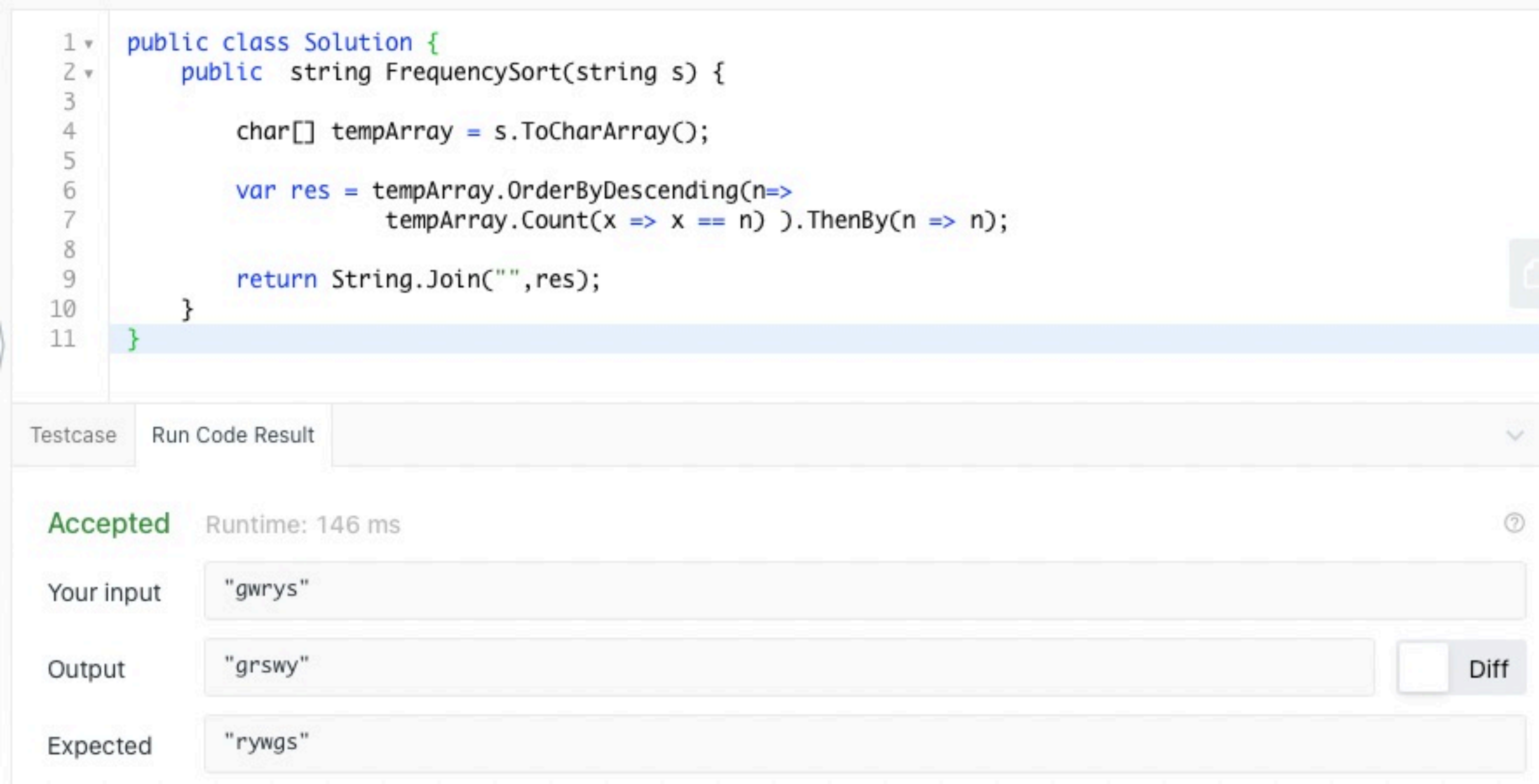
```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                    tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Accepted Runtime: 130 ms

Your input "bldit"

Output "bdilt"

Expected "blitd"



```
public class Solution {
    public  string FrequencySort(string s) {

        char[] tempArray = s.ToCharArray();

        var res = tempArray.OrderByDescending(n=>
                   tempArray.Count(x => x == n) ).ThenBy(n => n);

        return String.Join("",res);
    }
}
```

Testcase   Run Code Result

Accepted   Runtime: 146 ms

Your input   "gwrys"

Output   "grswy"   Diff

Expected   "rywgs"