Main: It manages to all operations. Reading file, taking integer arrays from taken string, processing these arrays and writing output of each operation to file and terminal is parts of main function.

Read: It reads the input file and store it to memory without changing it.

OpenFileToWrite : It opens the output file to save all outputs. The file will be stay open until the program is about to end.

ConvertIntegerArrays: Read function saved input as it reads it. So, our integer values are represented as string. This function takes the readed string and it converts it to bunch of integer arrays. It puts –1 at the end of every array and \0 to end of last array, so that program can determine if there is remaining array or not. After these function, our input is ready to be processed to create output requested.

ArrayProcessing: It starts processing of arrays.

ProgramLoop: This function works for every given array and it has functions processes the taken array.

SaveArrayToProcess: It takes the array which is about to be processed and saves it to the process location.

GiveOutputForCurrentArray: This is the function which creates output from taken array. The program which finds and prints the longest increasing sequence in the given array is part of this function. It does its job for all arrays so it is in the programLoop.After this function is end program loop checks if it was the last loop or not. If it is, the loop does not start again. If it is not, there is/are still array(s) to process, so loop starts again.

This function consist of 2 nested loops to find wanted sequence for given array. How this function finds the sequence is in the pseudocode section.

fill_lengths_with_one: This functions fills the lengths array with one. This array is used to determine longest increasing sequence.

WriteIntToFile: This function takes an integer value, and writes it to the file. Because of MIPS assembly writes/reads strings to/from file, first the integer should be converted to the string. Integers with different number of digits should have been written to the file. (Maximum 3 digit, because we used byte to hold integers.)

Exit: inte This function close the output file and ends the program.

Time Complexity: $\Theta(n^2)$

Space Complexity: $\Theta(n)$

```
func(A[0....n-1],size)
    for i = 0 to n-1 do:
        lengths[i] = 1
    endfor
    sizeOfMaxFoundSequence = 1
    for i = 1 to n-1 do:
        for j = 0 to i-1 do:
            // A[i] comes after than A[j], if A[i] greater than A[j]
            // and A[j] have more or equal number of elements which is lower than it,
            // those numbers are also lower than A[i] too.
            if A[i] > A[j] AND lengths[i] <= lengths[j]
                lenghts[i] = lengths[j] + 1
            endif
        endfor
        if lengths[i] > sizeOfMaxFoundSequence
            sizeOfMaxFoundSequence = lengths[i]
        endif
    endfor
    // Size is found, find the sequence which has found size
    longestSequence = int [sizeOfMaxFoundSequence]
    temp = sizeOfMaxFoundSequence
    for i = size-1 to -1 do:
        // find the element which can be tempth element for requested sequence
        if lengths[i] == temp
            temp--
            longestSequence[temp] = A[i]
        endif
    endfor
    return longestSequence and sizeOfMaxFoundSequence
Endfunction
```

Explanation of Pseudocode

Algorithm starts with filling the lenghts array with all ones. All indexes of this array will hold the number of elements (plus 1, because initial values are one not zero) which is lower than the element which is in the main array's same index.

Example:

Main Array: 4,2,3,5

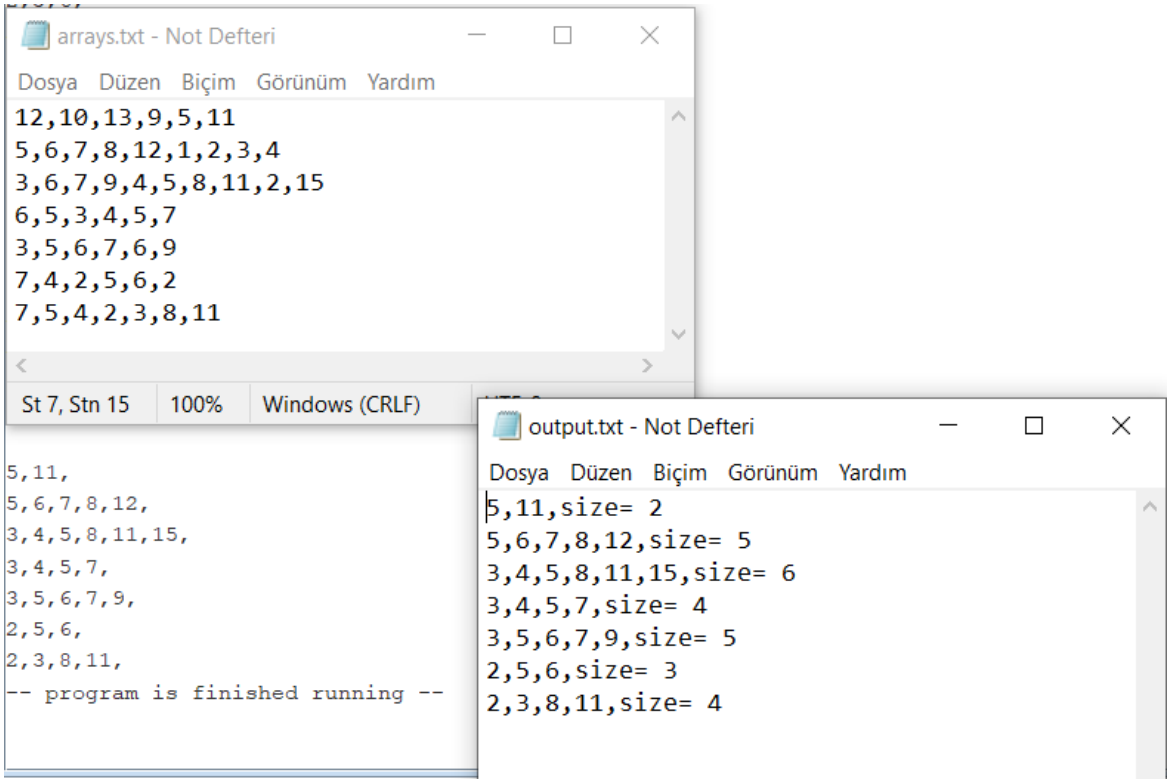
Lengths : 1,1,2,3

And these two nested loops, for every element between A[1] to A[n-1], it records how many elements exist which is less than the current element and have more or equal number of elements smaller than it. If it finds such elements, it assigns length of found element +1 to its lengths. Because it is bigger than the found element and its index is bigger. Current element is bigger than found element and all other elements less than found element.

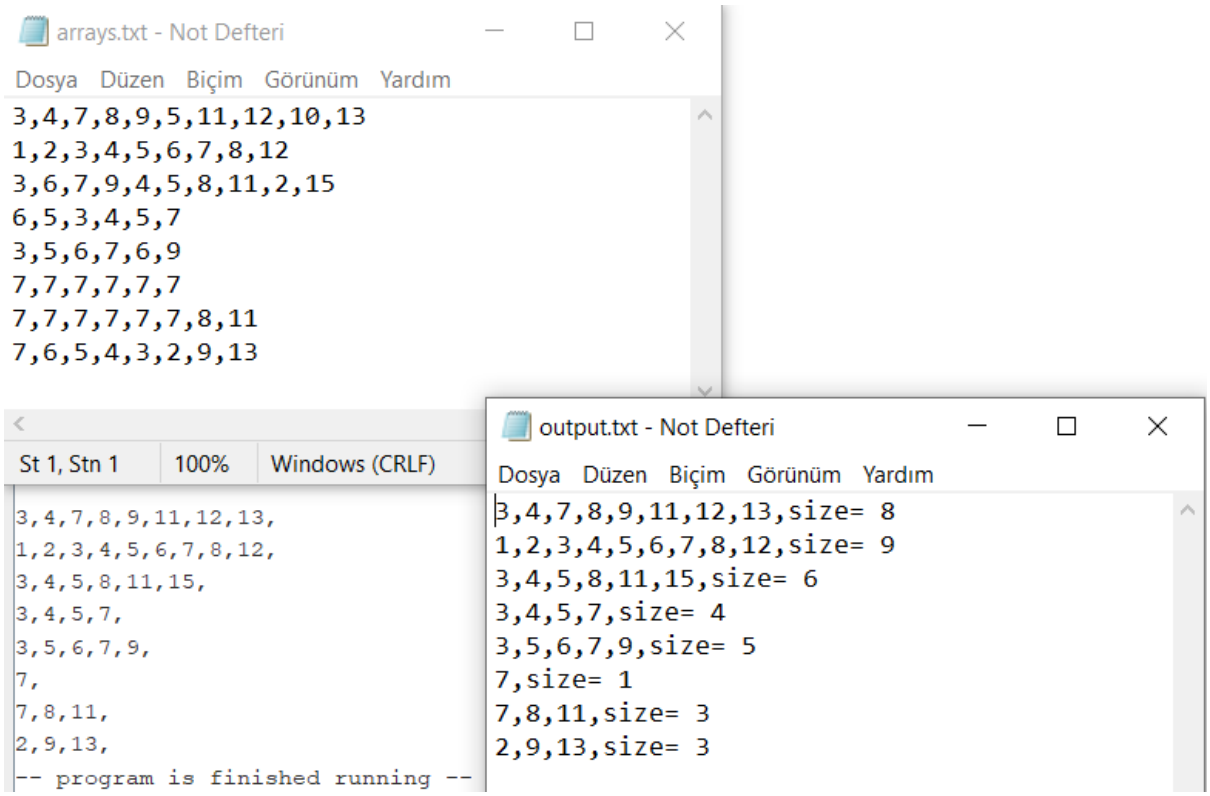
Maximum size is found with the help of the lenghts array.

Starting from the end of lenghts array, until the zeroth index ,for every element , it checks if it can be the biggest element in remaining part of the wanted sequence. If it is, it takes it and saves it to correct index of wanted sequence.

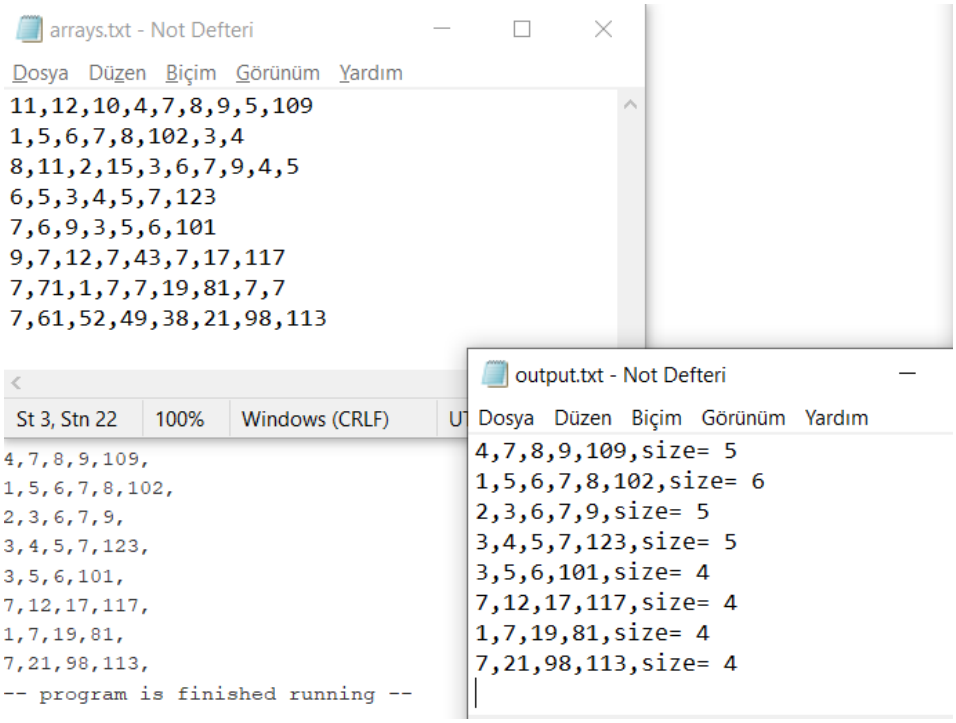
Test #1:



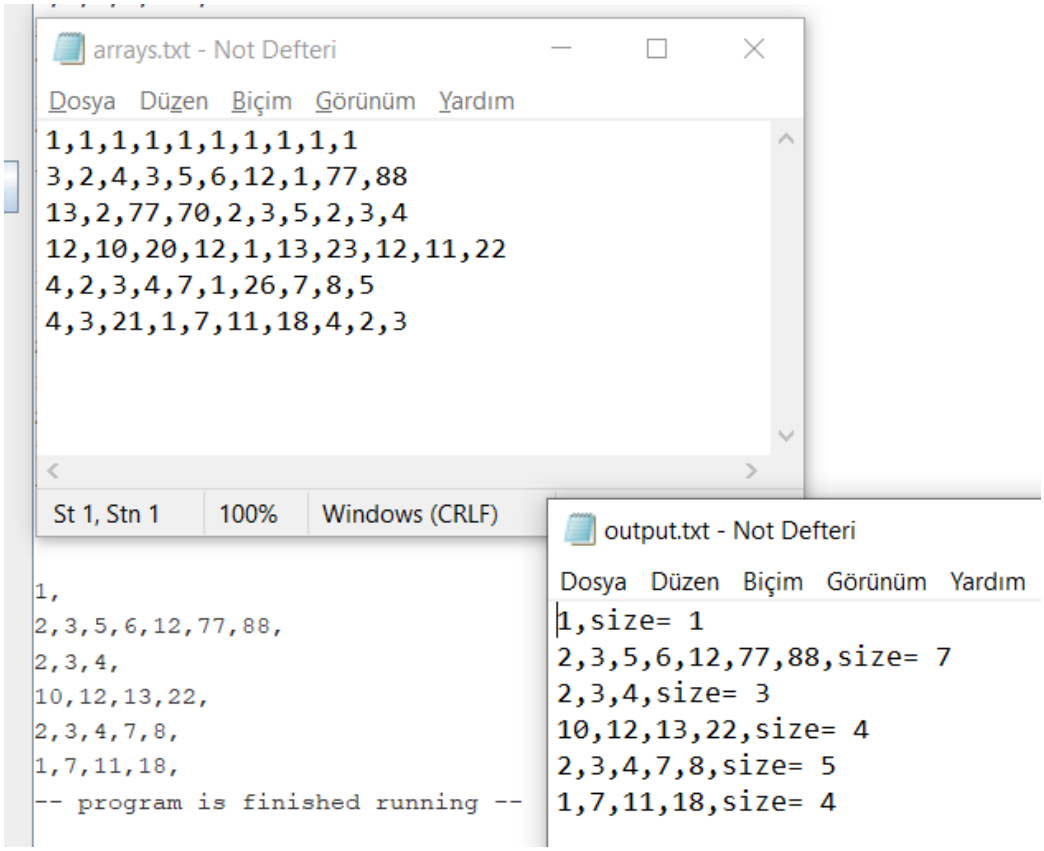
Test #2:


arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
3,4,7,8,9,5,11,12,10,13
1,2,3,4,5,6,7,8,12
3,6,7,9,4,5,8,11,2,15
6,5,3,4,5,7
3,5,6,7,6,9
7,7,7,7,7,7
7,7,7,7,7,7,8,11
7,6,5,4,3,2,9,13
St 1, Stn 1
100%
Windows (CRLF)
3,4,7,8,9,11,12,13,
1,2,3,4,5,6,7,8,12,
3,4,5,8,11,15,
3,4,5,7,
3,5,6,7,9,
7,
7,8,11,
2,9,13,
-- program is finished running --
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
3,4,7,8,9,11,12,13,size= 8
1,2,3,4,5,6,7,8,12,size= 9
3,4,5,8,11,15,size= 6
3,4,5,7,size= 4
3,5,6,7,9,size= 5
7,size= 1
7,8,11,size= 3
2,9,13,size= 3

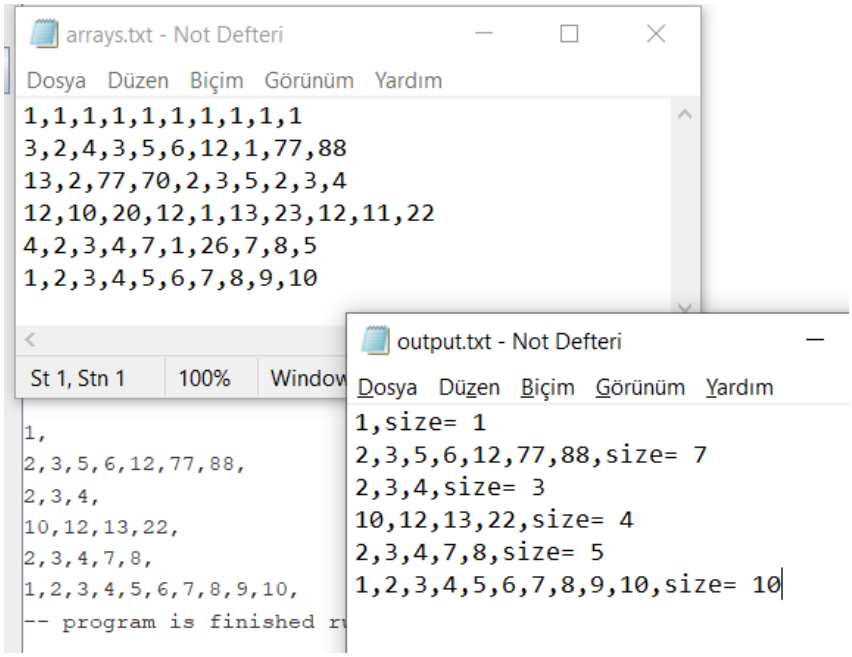
Test #3:


arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
11,12,10,4,7,8,9,5,109
1,5,6,7,8,102,3,4
8,11,2,15,3,6,7,9,4,5
6,5,3,4,5,7,123
7,6,9,3,5,6,101
9,7,12,7,43,7,17,117
7,71,1,7,7,19,81,7,7
7,61,52,49,38,21,98,113
St 3, Stn 22
100%
Windows (CRLF)
4,7,8,9,109,
1,5,6,7,8,102,
2,3,6,7,9,
3,4,5,7,123,
3,5,6,101,
7,12,17,117,
1,7,19,81,
7,21,98,113,
-- program is finished running --
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
4,7,8,9,109,size= 5
1,5,6,7,8,102,size= 6
2,3,6,7,9,size= 5
3,4,5,7,123,size= 5
3,5,6,101,size= 4
7,12,17,117,size= 4
1,7,19,81,size= 4
7,21,98,113,size= 4

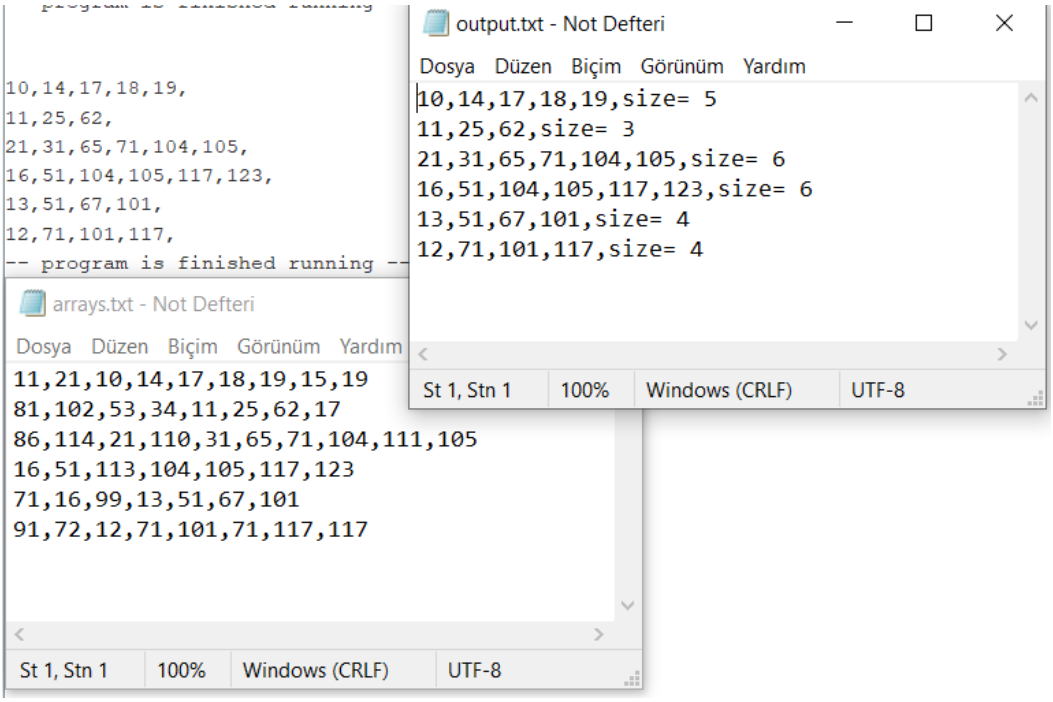
Test #4:


arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
1,1,1,1,1,1,1,1,1,1
3,2,4,3,5,6,12,1,77,88
13,2,77,70,2,3,5,2,3,4
12,10,20,12,1,13,23,12,11,22
4,2,3,4,7,1,26,7,8,5
4,3,21,1,7,11,18,4,2,3
St 1, Stn 1
100%
Windows (CRLF)
1,
2,3,5,6,12,77,88,
2,3,4,
10,12,13,22,
2,3,4,7,8,
1,7,11,18,
-- program is finished running --
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
1,size= 1
2,3,5,6,12,77,88,size= 7
2,3,4,size= 3
10,12,13,22,size= 4
2,3,4,7,8,size= 5
1,7,11,18,size= 4

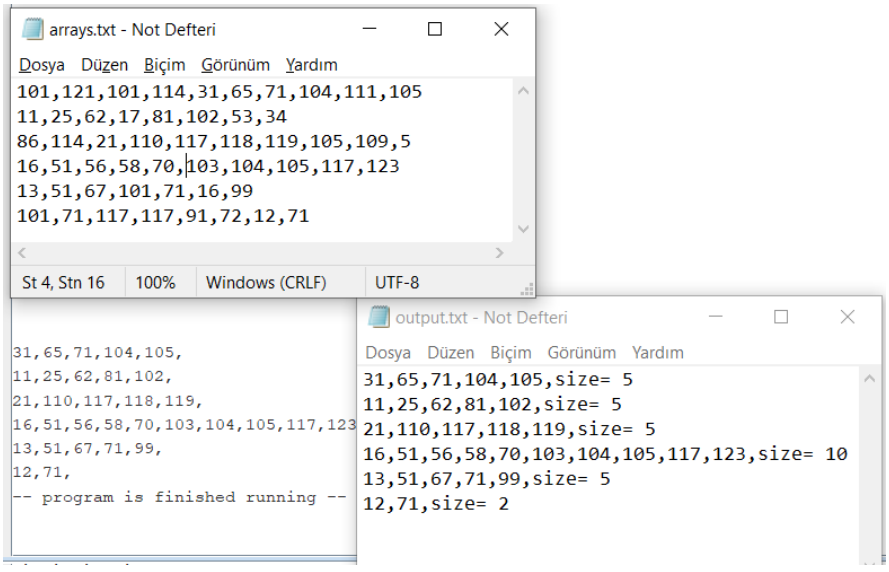
Test #5


arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
1,1,1,1,1,1,1,1,1,1
3,2,4,3,5,6,12,1,77,88
13,2,77,70,2,3,5,2,3,4
12,10,20,12,1,13,23,12,11,22
4,2,3,4,7,1,26,7,8,5
1,2,3,4,5,6,7,8,9,10
St 1, Stn 1
100%
1,
2,3,5,6,12,77,88,
2,3,4,
10,12,13,22,
2,3,4,7,8,
1,2,3,4,5,6,7,8,9,10,
-- program is finished
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
1,size= 1
2,3,5,6,12,77,88,size= 7
2,3,4,size= 3
10,12,13,22,size= 4
2,3,4,7,8,size= 5
1,2,3,4,5,6,7,8,9,10,size= 10

Test #6:


10,14,17,18,19,
11,25,62,
21,31,65,71,104,105,
16,51,104,105,117,123,
13,51,67,101,
12,71,101,117,
-- program is finished running --
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
10,14,17,18,19,size= 5
11,25,62,size= 3
21,31,65,71,104,105,size= 6
16,51,104,105,117,123,size= 6
13,51,67,101,size= 4
12,71,101,117,size= 4
St 1, Stn 1
100%
Windows (CRLF)
UTF-8
arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
11,21,10,14,17,18,19,15,19
81,102,53,34,11,25,62,17
86,114,21,110,31,65,71,104,111,105
16,51,113,104,105,117,123
71,16,99,13,51,67,101
91,72,12,71,101,71,117,117
St 1, Stn 1
100%
Windows (CRLF)
UTF-8

Test #7:


arrays.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
101,121,101,114,31,65,71,104,111,105
11,25,62,17,81,102,53,34
86,114,21,110,117,118,119,105,109,5
16,51,56,58,70,103,104,105,117,123
13,51,67,101,71,16,99
101,71,117,117,91,72,12,71
St 4, Stn 16
100%
Windows (CRLF)
UTF-8
31,65,71,104,105,
11,25,62,81,102,
21,110,117,118,119,
16,51,56,58,70,103,104,105,117,123
13,51,67,71,99,
12,71,
-- program is finished running --
output.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
31,65,71,104,105,size= 5
11,25,62,81,102,size= 5
21,110,117,118,119,size= 5
16,51,56,58,70,103,104,105,117,123,size= 10
13,51,67,71,99,size= 5
12,71,size= 2