ACSL 2022-23, Contest 2, Intermediate Divi... › hoonjung06@gmail.com

You can view this report online at : https://www.hackerrank.com/x/tests/1430699/candidates/50191344/report

Email:	hoonjung06@gmail.com
Test Name:	**ACSL 2022-23, Contest 2, Intermediate Division Programming Problem**
Taken On:	14 Feb 2023 16:45:23 PST
Time Taken:	27 min 41 sec/ 4320 min
Invited by:	ACSL Contests
Invited on:	14 Feb 2023 15:30:42 PST
Skills Score:	
Tags Score:	

100%

5/5

scored in **ACSL 2022-23, Contest 2, Intermediate Division Programming Problem** in 27 min 41 sec on 14 Feb 2023 16:45:23 PST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	**Binary Counting** › **Coding**	26 min 50 sec	5/ 5	✓

QUESTION 1

✓

Correct Answer

Score 5

Binary Counting › Coding

QUESTION DESCRIPTION

PROBLEM STATEMENT:

Given a string of characters found on the keyboard, convert each character in the string to the binary equivalent of its ASCII code. In the resulting concatenated string, search for the increasing sequence of binary numbers starting with 0, 1, 10, 11, … until a number cannot be found anywhere in the string. Look from the start of the string. If the binary number is found, remove that occurrence of the binary number from the string. Then look from the end of the string. If the binary number is found, remove that occurrence of the binary number from the string. When the binary number cannot be found at all, output the decimal equivalent of the last binary number that can be found.

EXAMPLE:

For the string “Roses are red.”, convert it to a concatenated string of binary numbers using each character’s ASCII code as follows:

Char	ASCII	Binary		Char	ASCII	Binary
R	82	01010010		r	114	01110010
o	111	01101111		e	101	01100101
s	115	01110011		sp	32	00100000

e	101	01100101		r	114	01110010
s	115	01110011		e	101	01100101
sp	32	00100000		d	100	01100100
a	97	01100001		.	46	00101110

Now search for binary numbers beginning with 0 in the following string:
01010010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 00101110

Remove the 0 from both ends so the string becomes:
1010010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 0010111
Remove the 1 from both ends so the string becomes:
010010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 001011
Remove 10 from both ends so the string becomes:
0010 01101111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 0011
Remove 11 from both ends so the string becomes:
0010 001111 01110011 01100101 01110011 00100000 01100001
01110010 01100101 00100000 01110010 01100101 01100100 00

Continuing until we search for 1001, the resulting string is:
000011010101110011000001100001011100100110010000000001000100
The string 1010 can only be found once from the start of the string so the resulting string is:
0000110111001100000110000101110010011001000000000 1000100
The process continues until the final string becomes:
000011000001100001001001000000000 1000100

TASK:

Complete the function **findLastBinary** that is called from a program that inputs the following data as its parameters and outputs the following information for each individual input:

- The function has one parameter: a string, *s*, that will be converted to binary using each character's ASCII code
- The function should return an integer representing the decimal equivalent to the last binary number that can be found in the string after all deletions have been made

You may create additional functions that are called from **findLastBinary** if needed in solving the problem.

CONSTRAINTS:

The input string may contain any character on the keyboard. The string will be fewer than 200 characters.

DATA PROVIDED:

There are 5 sets of Sample Data for debugging and 5 sets of Test Data for scoring. You may create additional data sets for debugging your program.

CANDIDATE ANSWER

Language used: **Python 3**

```
#
# Complete the 'findLastBinary' function below.
#
# The function is expected to return an INTEGER.
# The function accepts STRING s as parameter
```

```python
# The function accepts STRING s as parameter.
#

def findLastBinary(s):
    string = ""
    keepgoing = True
    c = 0
    bincounter = 0
    for l in s:
        string += bin(ord(l))[2::].zfill(8)

    while keepgoing:
        if str(bin(bincounter)[2::]) in string:
            string = string.replace(str(bin(bincounter)[2::]), "", 1)
            c+=1
        string = string[::-1]
        if str(bin(bincounter)[2::])[::-1] in string:
            string = string.replace(str(bin(bincounter)[2::])[::-1], "", 1)
            c+=1
        string=string[::-1]
        if c == 0:
            keepgoing = False
            break

        c=0
        bincounter+=1

    return bincounter-1

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	0	0.0581 sec	9.31 KB
Testcase 1	Easy	Sample case	✓ Success	0	0.0918 sec	9.26 KB
Testcase 2	Medium	Sample case	✓ Success	0	0.0587 sec	9.16 KB
Testcase 3	Medium	Sample case	✓ Success	0	0.0612 sec	9.36 KB
Testcase 4	Hard	Sample case	✓ Success	0	0.0548 sec	9.41 KB
Testcase 5	Easy	Hidden case	✓ Success	1	0.1025 sec	9.35 KB
Testcase 6	Medium	Hidden case	✓ Success	1	0.0915 sec	9.27 KB
Testcase 7	Medium	Hidden case	✓ Success	1	0.1107 sec	9.39 KB
Testcase 8	Hard	Hidden case	✓ Success	1	0.116 sec	9.34 KB
Testcase 9	Hard	Hidden case	✓ Success	1	0.0916 sec	9.35 KB

No Comments