

Coding Arena

Change Default Language ▾

Time Left

05 59 21
 hr min sec

A B C D E F G H

Problem : RLE String

Given a Run Length Encoded ASCII (Base 64) String as input, write a program to decode the string and print the corresponding image. Number of characters in each row of the resulting image is always 57.

This question can be understood once we understand how RLE maps to an image. Have a look at the Example section to get this understanding.

Input Format:

RLE ASCII String

Output Format:

Ascii data resembling an image corresponding to input RLE String

Example:

The following string is a RLE string

DJJGKHG HBMBEBIBFBF HBMBQDK HBMBTDH HBMBEBIBFBF HBNGKHG

It gets converted into the image below. You should see 'TCS' emerge by embossing the '!' character on '0' character

```

0000!!!!!!!!!!0000000000!!!!!!!!!!0000000000!!!!!!!!!!00000000
00000000!!00000000000000!!00000!!000000000!!000000!!000000
00000000!!00000000000000!!000000000000000000!!000000000000
00000000!!00000000000000!!0000000000000000000000!!000000000
00000000!!00000000000000!!00000!!000000000!!000000!!000000
00000000!!00000000000000!!!!!!!!!!0000000000!!!!!!!!!!00000000
  
```

Fig. 1. Ascii Image corresponding to RLE string shown above

Note: - Each row comprising only of '!' and '0' has a length of exactly 57 characters

Now let's understand the steps to convert the RLE into the image

- Split the RLE string on space character. For each substring achieved after splitting do step 2
- Lets perform the required calculation on first row of ascii image:-** which is **DJJGKHG**
 - Where, highlighted value is calculated as (Ascii value of(D)-64)if "" (68-64)=**4**
 - (Ascii value of(J)-64)-> (74-64)=**10**
 - (Ascii value of(J)-64)-> (74-64)=**10**
 - (Ascii value of(G)-64)-> (71-64)=**7**
 - (Ascii value of(K)-64)-> (75-64)=**11**
 - (Ascii value of(H)-64)-> (72-64)=**8**
 - (Ascii value of(G)-64)-> (71-64)=**7**
- Addition of all numerical values for each character in every sub-string as computed in step 2 should equal 57
- Corresponding to each alphabet in the sub-string, say D, a number, 4 in this case, is achieved by operation denoted in step 2
- Print '0' 4 times (let's call this operation multiply) which will produce a string '0000'
- Take the next character, J in this case and number 10 achieved by performing operation denoted in step 2
- Toggle '0' to '!' and print it 10 times which will produce a string '!!!!!!!!!!'

[Rules & Regulations](#)[Launch Code Editor](#)[Notifications](#)[Status messages](#)

8. Similarly, for each even index in sub-string multiply '0' with corresponding number and for each odd index in sub-string multiply '!' with corresponding number. Sub-string starts with index 0.
9. After processing each sub-string you will get the corresponding row in the ascii image
10. When all sub-strings are processed you will get all rows which will form the ascii image. Print that as output.

Sample Input and Output

Input1:

DJJGKHG HBMBEBIBFBF HBMBQDK HBMBTDH HBMBEBIBFBF HBNGKHG

Output1:

```
0000!!!!!!!!0000000000!!!!!!!!0000000000!!!!!!!!000000
00000000!000000000000!00000!00000000!00000!000000
00000000!000000000000!000000000000000!0000000000
00000000!000000000000!0000000000000000!00000000
00000000!000000000000!0000!00000000!00000!000000
00000000!000000000000!!!!!!!!0000000000!!!!!!!!000000
```

Input2:

YFZ YFZ YFZ YFZ JdK JdK JdK YFZ YFZ YFZ YFZ YFZ

Output1:

```
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
0000000000!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!0000000000
0000000000!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!0000000000
0000000000!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!0000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
00000000000000000000000000!!!!!!!!000000000000000000000000
```

Note:

Please do not use package and namespace in your code. For object oriented languages your code should be written in one class.

Note:

Participants submitting solutions in C language should not use functions from <conio.h> / <process.h> as these files do not exist in gcc

Note:

For C and C++, return type of main() function should be int.

© 2017 Tata Consultancy Services Limited. All Rights Reserved.

Submit Answer

- ☒ I, **ABHISHEK SARKAR** confirm that the answer submitted is my own.
- ☐ I would like to provide attribution to the following sources.

Select Language ▼

Select File

Browse...

Submit



