

Started on Tuesday, 7 October 2025, 8:24 PM

State Finished

Completed on Tuesday, 7 October 2025, 9:52 PM

Time taken 1 hour 28 mins

Marks 5.00/5.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Given a number, convert it into corresponding alphabet.

Input	Output
1	A
26	Z
27	AA
676	YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

$1 \leq \text{num} \leq 4294967295$

Sample Input 1

26

Sample Output 1

Z

For example:

Test	Result
print(excelNumber(26))	Z

Answer: (penalty regime: 0 %)

Reset answer

```
1 def excelNumber(n):
2     result=""
3     while n>0:
4         n-=1
5         result=chr(n%26+ ord('A')) + result
6         n//=26
7     return result
```

Test	Expected	Got	
✓ print(excelNumber(26))	Z	Z	✓
✓ print(excelNumber(27))	AA	AA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

The program must accept a string S which represents a series of keystrokes as the input. There are two special keystrokes denoted by "^z" for undo operation and "^y" for redo operation. The undo operation removes the last typed character and redo operations reverses the immediately done last undo operations. The program must accept the keystrokes and perform all the operations (undo and redo). Finally, the program must print the string after applying all the operations as the output.

Boundary Condition(s):

1 <= Length of string <= 1000

Input Format:

The first line contains the string S.

Output Format:

The first line contains the modified string.

Example Input/Output 1:

Input:

missel^z^z^z^ypell^z^z^yt

Output:

misspelt

Explanation:

The characters **missel** are followed by three undo operations which deletes last three characters.

So the string **missel** becomes **mis**.

The three undo operations are followed by one redo operation which reverses the last undo operation.

So the string **mis** becomes **miss**.

Then the characters **pell** are entered which are followed by two undo operations so the last two characters are removed.

So the string becomes **misspe**.

Then a redo operation is applied which reverses the last undo operation and so the string **misspe** becomes **misspel**.

The characters are followed by **t** so the final string is misspelt.

Answer: (penalty regime: 0 %)

```
1 s=input()
2 n=len(s)
3 l=[ ]
```

```
4 k=[]
5 i=0
6 while(i<n):
7     if(s[i:i+2]=='^z'):
8         if(l):
9             k.append(l.pop())
10        i+=2
11    elif s[i:i+2]=='^y':
12        if k:
13            l.append(k.pop())
14        i=i+2
15    else:
16        l.append(s[i])
17        i+=1
18 print(''.join(l))
19
```

	Input	Expected	Got	
✓	missel^z^z^z^ypell^z^z^yt	misspelt	misspelt	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input:

is2 sentence4 This1 a3

Output:

This is a sentence

Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input:

Myself2 Me1 I4 and3

Output:

Me Myself and I

Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

$2 \leq s.length \leq 200$

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in s is between 1 and 9.

The words in s are separated by a single space.

s contains no leading or trailing spaces.

Answer: (penalty regime: 0 %)

```

1 s=input().split()
2 r=[""]*len(s)
3 for word in s:
4     po=int(word[-1])-1
5     r[po]=word[:-1]
6 print(" ".join(r))

```

	Input	Expected	Got	
✓	is2 sentence4 This1 a3	This is a sentence	This is a sentence	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output:

1

Explanation: We cannot type "world" because the 'd' key is broken.

For example:

Input	Result
hello world ad	1
Faculty Upskilling in Python Programming ak	2

Answer: (penalty regime: 0 %)

```

1 w=input()
2 w=w.replace(' ','')
3 w=str(set(w))
4 b=set(input())
5 c=0
6 for ch in w:
7     if ch in b:
8         c+=1
9 print(c)

```

	Input	Expected	Got	
✓	hello world ad	1	1	✓
✓	Welcome to REC e	1	1	✓

	Input	Expected	Got	
✓	Faculty Upskilling in Python Programming ak	2	2	✓

Passed all tests! ✓

Correct

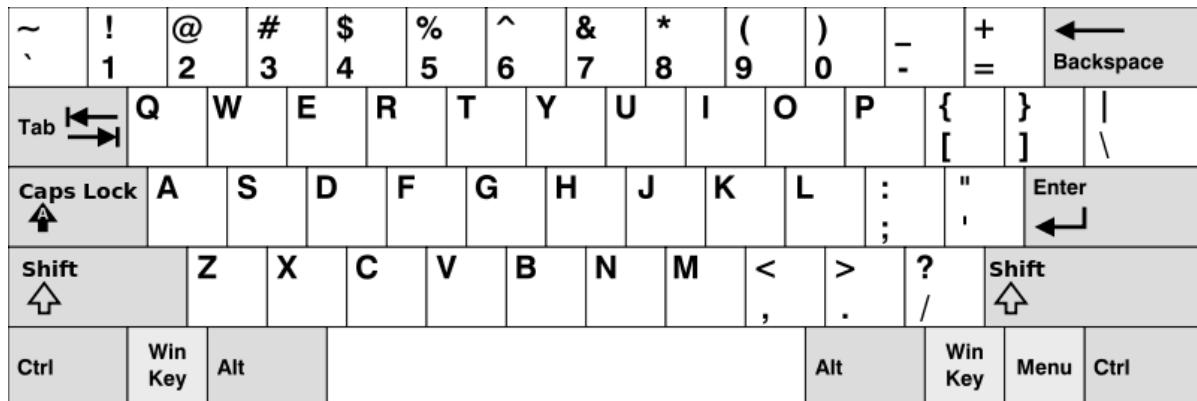
Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Given an array of strings `words`, return *the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below.*

In the **American keyboard**:

- the first row consists of the characters "qwertyuiop",
- the second row consists of the characters "asdfghjkl", and
- the third row consists of the characters "zxcvbnm".

**Example 1:**

```
Input: words = ["Hello", "Alaska", "Dad", "Peace"]
```

```
Output: ["Alaska", "Dad"]
```

Example 2:

```
Input: words = ["omk"]
```

```
Output: []
```

Example 3:

```
Input: words = ["adsdf", "sfd"]
```

```
Output: ["adsdf", "sfd"]
```

For example:

Input	Result
4	Alaska
Hello	Dad
Alaska	
Dad	
Peace	
2	adsdf
adsfd	afd
afd	

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 l=[]
3 for i in range(n):
```

```

4     l.append(input())
5     s1="qwertyuiop"
6     s2="asdfghjkl"
7     s3="zxcvbnm"
8     result=[]
9     for i in l:
10        w=i.lower()
11        if all(c in s1 for c in w) or all(c in s2 for c in w) or all(c in s3 for c in w):
12            result.append(i)
13    if result:
14        for i in result:
15            print(i)
16    else:
17        print("No words")

```

	Input	Expected	Got	
✓	4 Hello Alaska Dad Peace	Alaska Dad	Alaska Dad	✓
✓	1 omk	No words	No words	✓
✓	2 adsfd afd	adsfd afd	adsfd afd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.