| | |
|---|---|
| **Started on** | Tuesday, 7 October 2025, 12:03 PM |
| **State** | Finished |
| **Completed on** | Tuesday, 7 October 2025, 9:55 PM |
| **Time taken** | 9 hours 52 mins |
| **Marks** | 5.00/5.00 |
| **Grade** | **100.00** out of 100.00 |

## Question 1

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 <= s.length <= 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 %)

```
s=input()
w=s.split()
r=[""]*len(s)
for i in w:
    po=int(i[-1])-1
    r[po]=i[:-1]
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 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 %)

```
s=input()
n=len(s)
l=[]
k=[]
i=0
while(i<n):
    if s[i:i+2]=='^z':
        if l:
            k.append(l.pop())
        i+=2
    elif s[i:i+2]=="^y":
        l.append(k.pop())
        i+=2
    else:
        l.append(s[i])
        i+=1
print(''.join(l))
```

| | 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.

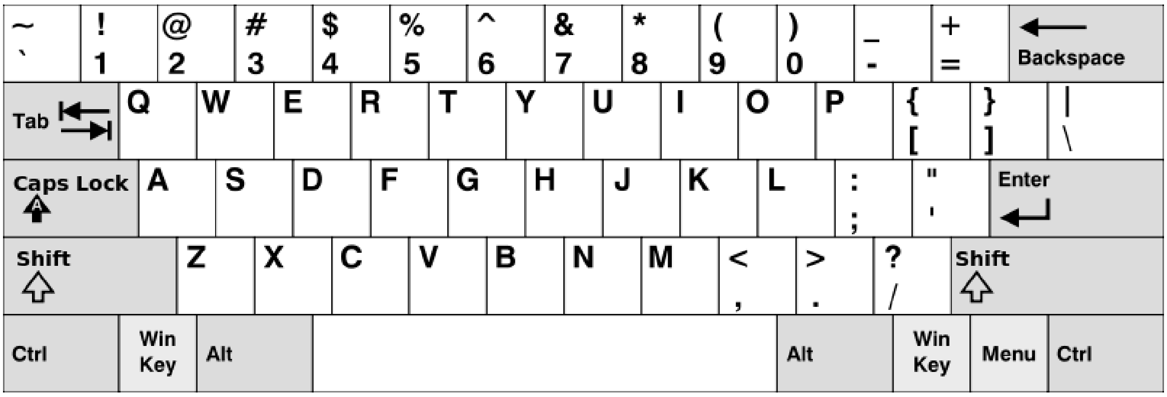
## Question 3

Correct Mark 1.00 out of 1.00

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"`.


~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 ( 9 ) 0 _ - + = Backspace
Tab Q W E R T Y U I O P { [ } ] | \
Caps Lock A S D F G H J K L : ; " ' Enter
Shift Z X C V B N M < , > . ? / Shift
Ctrl Win Key Alt Alt Win Key Menu Ctrl

**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<br>Hello<br>Alaska<br>Dad<br>Peace | Alaska<br>Dad |
| 2<br>adsfd<br>afd | adsfd<br>afd |

**Answer:** (penalty regime: 0 %)

```
n=int(input())
l=[]
for i in range(n):
    l.append(input())
s1='qwertyuiop'
s2='asdfghjkl'
s3='zxcvbnm'
result=[]
for i in l:
    w=i.lower()
    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):
        result.append(i)
if result:
    for i in result:
        print(i)
else:
    print('No words')
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | 4<br>Hello<br>Alaska<br>Dad<br>Peace | Alaska<br>Dad | Alaska<br>Dad | ✓ |
| ✓ | 1<br>omk | No words | No words | ✓ |
| ✓ | 2<br>adsfd<br>afd | adsfd<br>afd | adsfd<br>afd | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

## Question 4

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 <= num <= 4294967295

**Sample Input 1**

26

**Sample Output 1**

Z

**For example:**

| Test | Result |
|---|---|
| `print(excelNumber(26))` | Z |

**Answer:** (penalty regime: 0 %)

Reset answer

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

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<br>ad | 1 |
| Faculty Upskilling in Python Programming<br>ak | 2 |

**Answer:** (penalty regime: 0 %)

```
w=input()
w=w.replace(' ','')
w=str(set(w))
b=set(input())
c=0
for i in w:
    if i in b:
        c=c+1
print(c)
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | hello world<br>ad | 1 | 1 | ✓ |
| ✓ | Welcome to REC<br>e | 1 | 1 | ✓ |
| ✓ | Faculty Upskilling in Python Programming<br>ak | 2 | 2 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.