

Question-1:

Duplicate Attendance

XYZ College of Technology conducted a workshop on “Machine learning” in an auditorium. It is a big auditorium with 150 as its maximum capacity. Faculties felt difficult to take attendance, so they planned to circulate n attendance sheets to the students to mark their attendance. But few students unknowingly marked attendance more than one time in a sheet. Now it is difficult for the faculty to find the duplicate entries of a student.

Help the faculty to remove the duplicate attendance entry.

Note: Use **tuple** to solve the above problem

Input and Output Format:

First line of input consists of an integer n , which corresponds to the number of sheet.

Next n set of input correspond to students register number

Display Initial Attendance sheet with/without duplicate register numbers. Each inner tuple is considered as a single attendance sheet.

Display the unique register numbers.

Refer sample input and output for formatting specifications.

[All text in bold corresponds to the input and the rest corresponds to the output.]

Sample Input and Output:

Enter total Number of sheets:

6

1

2 3 1

4 5 6

2 3 4 5 6 7

2

3 4

Attendance Sheets with Register Number: ((1,), (2, 3, 1), (4, 5, 6), (2, 3, 4, 5, 6, 7), (2,), (3, 4))

Final sheet: (1, 2, 3, 4, 5, 6, 7)

Code:

```
n = int(input("Enter total Number of sheets:\n"))
```

```
attendance_sheets = []
```

```
for _ in range(n):
```

```
    sheet = tuple(map(int, input().split()))
```

```
attendance_sheets.append(sheet)
print("Attendance Sheets with Register Number:", tuple(attendance_sheets))
unique_registers = set()
for sheet in attendance_sheets:
    unique_registers.update(sheet)
final_sheet = tuple(sorted(unique_registers))
print("Final sheet:", final_sheet)
```

Question-2:

Nella's Birthday and Colorfull Balloons

Nella is an eight-year-old princess who will inherit the kingdom of Castlehaven. It is her birthday today and her Dad, the King of Castlehaven has arranged for a grand party. Nella loves colorful balloons and so her Dad planned to decorate the entire palace with balloons of the colors that Nella loved. So he asked her about her color preferences. The sophisticated princess, that is Nella, likes only two colors — amber and brass. Her Dad bought n balloons, each of which was either amber or brass in color.

You are provided this information in a string s consisting of characters 'a' and 'b' only, where 'a' denotes that the balloon is amber, where 'b' denotes it being brass colored. When Nella saw the balloons, she was furious with anger as she wanted all the balloons of the same color. In her rage, she painted some of the balloons with the opposite color (i.e., she painted some amber ones brass and vice versa) to make all balloons appear to be the same color.

It took her a lot of time to do this, but you can probably show her the right way of doing so, thereby teaching her a lesson to remain calm in difficult situations, by finding out the minimum number of balloons needed to be painted in order to make all of them the same color.

Input Format:

The first and only line of input contains a string s .

Output Format:

Output a single line containing an integer — the minimum number of flips required. Refer sample input and output for formatting specifications.

Sample Input1:

ab

Sample Output1:

1

Sample Input2:

baaba

Sample Output2:

2

Code:

```
s = input()
```

```
c = 0
```

```
d = 0
```

```
for char in s:
```

```
    if char == 'a':
```

```
        c += 1
```

```
    elif char == 'b':
```

```
        d += 1
```

```
    else:
```

```
        print("Enter string with a and b characters only")
```

```
        break
```

```
else:
```

```
    if c > d:
```

```
        print(d)
```

```
    elif c < d:
```

```
        print(c)
```

```
    else:
```

```
        print(c)
```

Question-3:

Spell Bee Contest

Given a pair of words (the first is the correct spelling and the second is the contestant's spelling of the word) determine if the word is spelt correctly.

The degree of correctness is as follows:

- CORRECT if it is an exact match
- ALMOST CORRECT if no more than 2 letters are wrong
- WRONG if 3 or more letters are wrong

Input Format:

Input consists of a pair of words.

Output Format:

Output the contestant's spelling of the word and the degree of correctness. All letters are upper case. The output is to be formatted exactly like that for the sample output given below.

Assumptions:

Words contain only upper case letters. The maximum word length is 10.

Sample Input:

SAMPLE
SIMPLE

Sample Output:

SAMPLE IS ALMOST CORRECT

Code:

```
s = input()
c = input()
co = 0

for i in range(min(len(s), len(c))):
    if c[i] != s[i]:
        co += 1

if co == 0:
    print(f'{c} IS CORRECT')
elif co in (1, 2):
    print(f'{c} IS ALMOST CORRECT')
else:
    print(f'{c} IS WRONG')
```

Question-4:

Reverse the digits

Asha's birthday is shortly coming and her parents have planned to arrange for a house party. Deepa was Asha's best friend and was expecting her birthday since a month. This is because Deepa's Dad has promised her that he and Deepa together would design a Reverse Talking kitty toy all by themselves and gift it to Asha. Deepa believed that Asha might be overjoyed with this gift from her dear friend.

Deepa's Dad put the best of his efforts to design the toy. As a first module in the design he intended to write a program that would reverse a numeric input given to it. He needs your help to write a recursive function for reversing the digits of the given number N. Please assist him in the task.

Python Function Specifications:

Use the function name and the argument as:

```
def reverse(n):
```

This recursive function should return the reverse of a N digit number.

C Function Specifications:

Use the function name and the argument as:

```
int reverse(n):
```

This recursive function should return the reverse of a N digit number.

Input Format:

The first line of the input is an integer N.

Output Format:

Print the reverse of a N digit number.

Refer sample input and output for formatting specifications.

Sample Input 1:

1234

Sample Output 1:

4321

Sample Input 2:

32333333

Sample Output 2:

33333323

Code:

```
def reverse(n):
    global sum
    if n:
        r = n % 10
        sum = sum * 10 + r
        reverse(n // 10)
    else:
        return 0
    return sum

if __name__ == "__main__":
    sum = 0
    n = int(input())
    print(reverse(n))
```

Question-5:

2094. Finding 3-Digit Even Numbers

You are given an integer array `digits`, where each element is a digit. The array may contain duplicates.

You need to find **all** the **unique** integers that follow the given requirements:

- The integer consists of the **concatenation** of **three** elements from `digits` in **any** arbitrary order.
- The integer does not have **leading zeros**.
- The integer is **even**.

For example, if the given `digits` were `[1, 2, 3]`, integers `132` and `312` follow the requirements.

Return a **sorted** array of the unique integers.

Example 1:

Input: digits = [2,1,3,0]

Output: [102,120,130,132,210,230,302,310,312,320]

Explanation: All the possible integers that follow the requirements are in the output array.

Notice that there are no **odd** integers or integers with **leading zeros**.

Example 2:

Input: digits = [2,2,8,8,2]

Output: [222,228,282,288,822,828,882]

Explanation: The same digit can be used as many times as it appears in digits.

In this example, the digit 8 is used twice each time in 288, 828, and 882.

Example 3:

Input: digits = [3,7,5]

Output: []

Explanation: No **even** integers can be formed using the given digits.

Constraints:

- $3 \leq \text{digits.length} \leq 100$
- $0 \leq \text{digits}[i] \leq 9$

Code:

```
from itertools import permutations
```

```
class Solution:
```

```
    def findEvenNumbers(self, digits: List[int]) -> List[int]:
```

```
        result = set()
```

```
        for perm in permutations(digits, 3):
```

```
            num = perm[0] * 100 + perm[1] * 10 + perm[2]
```

```
            if perm[0] != 0 and perm[2] % 2 == 0:
```

```
                result.add(num)
```

```
        return sorted(result)
```