**Code Test 1**

**2D String Array Sub-String Search**

Given a string S of length L, the program must store the characters of the string S in a two dimensional array and search for a given sub-string SUB in the two dimensional array both from left to right and from top to bottom. If found the program must print the start index and end index of the sub-string. (Note: The sub-string (if present) will be present only once in the 2D array). If the sub-string is NOT present the program must print -1 as the output.

**Input Format:**  
The first line contains the value of the string S.  
The second line contains the value of the sub-string SUB.  
The third line contains the column count in a given row of the 2D array.

**Boundary Conditions:**  
The length of the String S is from 4 to 100.  
4 <= L (Length of S) <= 100

**Output Format:**  
If the sub-string is found,  
The first line contains the start index.  
The second line contains the end index.

If the sub-string is not found,  
The first line contains -1

**Example Input/Output 1:**  
Input:  
WELCOMETOZOHOCORPORATION  
TOO  
5

Output:  
1,2  
3,2

Explanation:  
As there are 5 columns in a row in the 2D array, the representation is like  
W E L C O  
M E T O Z  
O H O C O  
R P O R A  
T I O N

TOO is found when searching from top to bottom in 3rd column (starting from 2nd row and ending in 4th row)  
So the start index is 1,2 (As index starts from 0)  
and the end index is 3,2

**Example Input/Output 2:**  
Input:  
THEPOTCALLINGKETTLEBLACK  
TTL  
6

Output:  
2,3  
2,5

Explanation:  
As there are 6 columns in a row in the 2D array, the representation is like  
T H E P O T  
C A L L I N  
G K E T T L  
E B L A C K

TTL is found in 3rd row, 4th column from left to right.  
So the start index is 2,3 (As index starts from 0)  
and the end index is 2,5

**Example Input/Output 3:**  
Input:  
MIDASTOUCHTURNSGOLD  
CAR  
4

Output:  
-1

Explanation:  
As there are 4 columns in a row in the 2D array, the representation is like  
M I D A  
S T O U  
C H T U  
R N S G  
O L D

CAR is not found traversing from left to right or top to bottom. So -1 is printed as the output.

**Code Test 2**

**Odd length string diagonal pattern**

Given a**string S of odd length L**, the program must print it twice as diagonals with the middle letter being the point of intersection.

**Input Format:**  
The first line contains the value of S.

**Boundary Conditions:**  
The length of the String S is from 3 to 20.  
1 <= L (Length of S) <= 20

**Output Format:**  
L lines printing the desired pattern.

**Example Input/Output 1:**  
Input:  
PROGRAM

Output:  
P     M  
 R   A  
  O R  
   G  
  O R  
 R   A  
P     M

**Example Input/Output 2:**  
Input:  
CABLE

Output:  
C   E  
 A L  
  B  
 A L  
C   E

**Code Test 3**

**Sort numbers based on weight**

Given a set of N numbers and the rules provided below to calculate their weights, the program must sort the numbers based on their weight and print the numbers in descending order.

**Rules to calculate weight:**  
    - 5 if a perfect square  
    - 4 if multiple of 4 and divisible by 6  
    - 3 if even number

**Input Format:**  
The first line contains the value of N.  
The next N lines contain the value of N numbers.

**Boundary Conditions:**  
The length of the array of numbers will be from 3 to 200.  
1 <= N <= 20

**Output Format:**  
N lines containing the sorted numbers based on their weight.

**Example Input/Output 1:**  
Input:  
5  
10  
36  
54  
49  
12

Output:  
36  
12  
49  
54  
10

Explanation:  
10's weight = 3 for just being an even number.  
36's weight = 5+4+3 = 12 (as it is a perfect square of 6, multiple of 4 and divisible by 6 and also it is an even number)  
54's weight = 3 for just being an even number  
49's weight = 5 (as it is a perfect square of 7)  
12's weight = 4+3 = 7 (multiple of 4 and divisible by 6 and also it is an even number)

In this 10 and 54 have same weight which is 3. Between them 54 is larger. So it is printed first.

**Example Input/Output 2:**  
Input:  
4  
89  
81  
72  
99

Output:  
72  
81  
99  
89

Explanation:  
89's weight = 0  
81's weight = 5 (for just being a perfect square)  
72's weight = 4+3 = 7 (multiple of 4 and divisible by 6 and also it is an even number)  
99's weight = 0

As 99 is greater than 89, 99 is printed first.