Wipro – Coding Questions

1)Print a given matrix in spiral form.

Given a 2D array, print it in spiral form. See the following examples.

Please comment down the code in other languages as well below –

```
Input:
        1    2   3   4
        5    6   7   8
        9   10  11  12
        13  14  15  16
Output:
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
```

2)To print the trapezium pattern?

Please also post your code in the comments in different languages or same languages with short or better time complexity code.

```
Input:

N = 4

Output:

1*2*3*4*17*18*19*20
  5*6*7*14*15*16
    8*9*12*13
      10*11

Input:

N = 5

Output:

1*2*3*4*5*26*27*28*29*30
  6*7*8*9*22*23*24*25
    10*11*12*19*20*21
      13*14*17*18
        15*16
```

3)Counting Valleys

Problem Statement

Gary is an avid hiker. He tracks his hikes meticulously, paying close attention to small details like topography. During his last hike, he took exactly n steps. For every step he took, he noted if it was an uphill or a downhill step. Gary’s hikes start and end at sea level. We define the following terms:

- A mountain is a non-empty sequence of consecutive steps above sea level, starting with a step up from sea level and ending with a step down to sea level.
- A valley is a non-empty sequence of consecutive steps below sea level, starting with a step down from sea level and ending with a step up to sea level.

Given Gary's sequence of up and down steps during his last hike, find and print the number of valleys he walked through.

Input Format :

- The first line contains an integer, , denoting the number of steps in Gary's hike.
- The second line contains a single string of characters. Each character belongs to {U, D} (where U indicates a step up and D indicates a step down), and the i(th) cin the string describes Gary's i(th) step during the hike.

Constraints :

- 2 <= N <= 10^6

Output Format

- Print a single integer denoting the number of valleys Gary walked through during his hike.

Input :

8

UDDDUDUU

Sample Output

1

Explanation

If we represent _ as sea level, a step up as / , and a step down as \ , Gary's hike can be drawn as:

```
_/\    _
  \   /
   \/\/
```

It's clear that there is only one valley there, so we print on a new line.

4)A Pythagorean triplet is a set of three integers a, b and c such that $a^2+b^2=c^2$. Given a limit, generate all Pythagorean Triples with values smaller than given limit.

```
Input :

     limit = 20

Output : 3 4 5
         8 6 10
         5 12 13
```

```
15 8 17
12 16 20
```

A Simple Solution is to generate these triplets smaller than given limit using three nested loop. For every triplet, check if Pythagorean condition is true, if true, then print the triplet. Time complexity of this solution is O(limit^3) where 'limit' is given limit.

An Efficient Solution can print all triplets in O(k) time where k is number of triplets printed. The idea is to use square sum relation of Pythagorean triplet, i.e., addition of squares of a and b is equal to square of c, we can write these number in terms of m and n such that,

$a = m^2 - n^2$
$b = 2 * m * n$
$c = m^2 + n^2$

because,

$a^2 = m^4 + n^4 – 2 * m^2 * n^2$
$b^2 = 4 * m^2 * n^2$
$c^2 = m^4 + n^4 + 2* m^2 * n^2$

We can see that $a^2 + b^2 = c^2$, so instead of iterating for a, b and c we can iterate for m and n and can generate these triplets

5) What will be the output of the following C code?

```
#include<stdio.h>
int main()
{
        int x=97;
        switch(x)
        {
        case 'a':
                printf("yes");
                break;
        case '97':
                printf("no");
                break;
        }
}
```

a) yes
b) yes no
c) Duplicate case value error
d) Character case value error

ANS: c

6)Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Ex 1:

```
Input: s = "()"
Output: true
```

Ex 2:

```
Input: s = "()[]{}"
Output: true
```

Ex 3:

```
Input: s = "(]"
Output: false
```

7)Longest Palindrome in a String

Given a string S, find the longest palindromic substring in S. Substring of string S: S[i . . . . j] where 0≤ i≤ j < len(S). Palindrome string: A string which reads the same backwards. More formally, S is palindrome if reverse(S) = S. Incase of conflict, return the substring which occurs first (with the least starting index).

Example 1:

```
Input:
S = "aaaabbaa"
Output: aabbaa
Explanation: The longest Palindromic
substring is "aabbaa".
```

Example 2:

```
Input:
S = "abc"
Output: a
Explanation: "a", "b" and "c" are the
longest palindromes with same length.
The result is the one with the least
starting index.
```

8)Largest number in K swaps
Given a number K and string str of digits denoting a positive integer, build the largest number possible by performing swap operations on the digits of str at most K times.

Example 1:

```
Input:
K = 4
str = "1234567"
Output:
7654321
Explanation:
Three swaps can make the
input 1234567 to 7654321, swapping 1
with 7, 2 with 6 and finally 3 with 5
```

Example 2:

```
Input:
K = 3
str = "3435335"
Output:
5543333
Explanation:
Three swaps can make the input
3435335 to 5543333, swapping 3
with 5, 4 with 5 and finally 3 with 4
```

9)Given a matrix of size NxM and a list of queries containing (a,b) pairs. Find the maximum sum among all (a x b) sub-matrices of the matrix.
Note: The rows and columns of the submatrix must be contiguous. For example:

Example 1:

```
Input:
N = 3, M = 4
mat[][] = {{1, 2, 3, 9},
           {4, 5, 6, 2},
           {8, 3, 2, 6}}
Q = 1
Queries[] = {(3,2)}

Output: 28

Explanation:
Here a = 3 and b = 2.

The first 3x2 submatrix is:
1 2
4 5
8 3
The sum of elements in this is 23.
```

```
The second 3x2 submatrix is:
2 3
5 6
3 2
The sum of elements in this is 21.

The third 3x2 submatrix is:
3 9
6 2
2 6
The sum of elements in this is 28.

The maximum among these is 28.
```

Example 2:

```
Input:
N = 3, M = 4
mat[][] = {{1, 2, 3, 9},
           {4, 5, 6, 2},
           {8, 3, 2, 6}}
Q = 3
Queries[] = {(1, 1), (2, 2), (3, 3)}

Output: 9 20 38
```

Example 3:

```
Input:
N = 2, M = 7
mat[][] = {{49, 44, 46, 22, 24, 11, 2},
           {5, 25, 24, 28, 20, 47, 47}}
Q = 1
Queries = {(2, 1)}

Output: 70
```

10)Course Schedule

There are a total of `numCourses` courses you have to take, labeled from `0` to `numCourses - 1`. You are given an array `prerequisites` where `prerequisites[i] = [a_i, b_i]` indicates that you must take course b_i first if you want to take course a_i.

- For example, the pair `[0, 1]`, indicates that to take course `0` you have to first take course `1`.

Return `true` if you can finish all courses. Otherwise, return `false`.

Example 1:

```
Input: numCourses = 2, prerequisites = [[1,0]]
Output: true
Explanation: There are a total of 2 courses to take.
To take course 1 you should have finished course 0. So it is
possible.
```

Example 2:

```
Input: numCourses = 2, prerequisites = [[1,0],[0,1]]
Output: false
Explanation: There are a total of 2 courses to take.
To take course 1 you should have finished course 0, and to take
course 0 you should also have finished course 1. So it is
impossible.
```