Question 1

Due on 2019-08-22, 23:59 IST
Level: Medium

In this assignment, you will be given an N x N matrix. You have to determine whether the matrix is a triangular matrix.

The diagonal of the matrix M of size N x N is the set of entries M(0,0), M(1,1), M(2,2), ..., M(N,N).

A matrix is upper triangular if every entry below the diagonal is 0. For example,
1 1 1
0 0 1
0 0 2
is an upper triangular matrix. (The diagonal itself, and the entries above and below the diagonals can be zeroes or non-zero integers.)

A matrix is lower triangular if every entry above the diagonal is 0. For example,
2 0 0
3 1 0
4 2 2
is a lower triangular matrix.

A matrix is triangular if it is either upper triangular or lower triangular or both.

You may not use arrays for this program.

Input
First, you will be given N, which is the size of the matrix.

Then you will be given N rows of integers, where each row consists of N integers separated by spaces.

Output
If the input matrix is triangular, then print yes. Otherwise, print no.

Sample Test Cases

Test Case 1

Input
3
1 0 0
0 1 0
1 1 2

Output
yes
## Introduction to Programming in C - Course

The due date for submitting this assignment has passed. As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

```c
#include<stdio.h>

int main()
{
  int i,x,p,q,n,counter,z;
  p=0;
  q=0;
  scanf("%d",&n);
  for(i=1;i<n+1;i++)
  {
    counter = 1;
    for(;counter<i;)
    {
      scanf("%d",&z);
      counter++;
      if(z==0);
    }
    if(counter==i)
    {
      scanf("%d",&x);
      counter++;
    }
    for(;counter<n+1;)
    {
      scanf("%d",&z);
      counter++;
      if(z==0);
    }
    else
    q=1;
  }
  if(p==1&q==1)
  printf("no");
  else
  printf("yes");
}
```

<table>
<thead>
<tr>
<th>Test Case 2</th>
<th>Test Case 3</th>
<th>Test Case 4</th>
</tr>
</thead>
</table>
| 7
1 0 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1 |
| yes | no | yes |
| 7
1 0 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 1
0 0 0 0 0 1 |
| 2
1 1
0 1 |
return 0;