*__Problem Statement 1__:
Given a square matrix A of dimension n*n, write a program rotate the matrix by 90 degrees in clockwise direction and print the final matrix. Use the following function prototype.

void rotateClkWise90(int A[][100], int n);

Constraints:

1<t<100

1<=n<=100

Input:
First Line consists value of ‘t’ denoting number of test cases.
In each case, first line denotes the value of ‘n’ which is the size of the matrix
Followed n lines with n elements in each row (nxn matrix).

Output:
Print the rotated matrix.

Sample Input:
1
3
1 2 3
8 9 4
7 6 5

Sample Output:
8 1 2
7 9 3
6 5 4

*Problem Statement 2:

Given two sorted arrays A and B, where the elements in A are in positions from low1 to high1, and the elements in B are in positions from low2 to high2, write a program to merge all the elements into an array C in sorted order. Use the following function prototype:

void merge(int A[], int B[], int C[], int low1, int high1, int low2, int high2);
Here high1 and high2 are the inputs.

Constraints:
1<t<100
0<high1,high2<=4999

Input:
First line gives the value of 't' which denotes number of test cases.
In each test case, First line consists of 'high1' and 'high2' seperated by spaces.
Second line consists 'high1' elements and third line consists 'high2' elements.

Output:
Single line containing all the elements of two input arrays in a sorted order seperated by spaces.

Sample Input:
1
3 4
1 3 5
2 4 6 8

Sample Output:
1 2 3 4 5 6 8

***Problem Statement 3:**

A big Integer and a single digit are given. Write a C Program to find their product. Use the following function prototype:

void multiply(char A[], char B[], int digit);

Constraints:

1<t<100

0<strlen(A)<9995

Input:

First line denotes number of test cases.

Next ‘t’ lines gives a string (big number) and a single digit integer seperated by space.

Output:

Product A*digit should be printed in a single line.

Sample Input:

1

111111111111111111111111111 9

Sample output:

999999999999999999999999999

*__Problem Statement 4: Smart – Reading__

Given a **char** array (say ‘ch’)for Charecter of a particular page and an **int** array for denoting the page number which user have to go. Write a program to print all the charecters in a smart order

Smart Order:

First print 0^{th} char and go to page number that 0^{th} page has and keep on traversing till you counter -1 as your next page number.

Use the following function prototype:

Void smartRead(char data[], int next[], char result[]);

Constraints:

1<t<100

0<strlen(ch)<9999

Input:

First line denotes number of test cases.

In each test case, first line consists of a char array of length ‘n’ (n is not an input).

second line consists of n integers seperated by spaces.

Output:

A single char array (without spaces) of defined order to be printed.

Sample Input:

2

@#$%A1`

5 6 2 1 -1 4 3

QwErTy

1 2 3 4 5 -1

Sample Output:

@1A

QwErTy

*__Problem Statement 5:__
Write a program to fill the upper triangle of 2D Matrix of order nxn with natural numbers in the following order:
First fill the elements in principle diagonal (from 1 to n) and the upper diagonal and so on..

Constraints:
1<t<100
0<n<99

Input:
First line denotes number of test cases.
Next t lines denotes the value of ‘n’.

Output:
nxn matrix in the given pattern.

Sample Input:
2
3
4

Sample Output:
1 4 6
0 2 5
0 0 3
1 5 8 10
0 2 6 9
0 0 3 7
0 0 0 4

Problem Statement 6:
Write a program to print the summation of middle two bytes of a given integer n.

Constraints:
1<t<100
1<n<9999

Input:
First line denotes number of test cases
Next lines consists of different values of n in each line.

Output:
Print sum of middle two bytes of respective input.

Sample Input:
2
4
257

Sample Output:
0
1

Problem Statement 7: Partition Problem

Note: **Will be Updated soon.....**