Assignment Solutions | 2D Arrays - 2 | Week 6

1.Write a program to print the elements of both the diagonals in a square matrix.
Input 1:
1 2 3
4 5 6
7 8 9
Output 1:
1 3
5
7 9
Solution :

```
#include<iostream>
using namespace std;
int main(){
int n ;
cout << "Enter the number of rows : ";
cin >> n;
int arr[n][n];
cout << "Enter the elements of matrix : "<<endl;
for(int i=0;i<n;i++){
for(int j=0;j<n;j++)cin>>arr[i][j];
}
cout << "Elements of both the diagonals are as follows : "<<endl;
for(int i = 0 ; i < n ; i++){
for(int j = 0 ; j < n ; j++){
if((i + j == n - 1) or (i == j))cout << arr[i][j] << " ";
else cout << " ";
}
cout<<endl;
}
}
```

2.Write a program to rotate the matrix by 90 degrees anti-clockwise.
Input 1:
1 2 3
4 5 6
7 8 9
Output 1:
3 6 9
2 5 8
1 4 7
Solution :

```
#include <bits/stdc++.h>
using namespace std;
int main() {
```

```
int n;
cin>>n;
int a[n][n];
for(int i=0;i<n;i++){
for(int j=0;j<n;j++)cin>>a[i][j];
}
// let's first calculate the transpose of the given matrix
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
if(i <= j)swap(a[i][j] , a[j][i]);
}
}
for(int j=0;j<n;j++){
for(int i=0;i<n/2;i++){
swap(a[i][j] , a[n-i-1][j]);
}
```

3.Write a program to print the matrix in wave form.
Input :
1 2 3
4 5 6
7 8 9
Output : 7 4 1 2 5 8 9 6 3
Solution :

```
#include<iostream>
using namespace std;
int main(){
int n , m;
cout << "Enter the number of rows : ";
cin >> n;
cout << "Enter the number of columns : ";
cin >> m;
int a[n][n];
cout << "Enter the matrix elements : "<<endl;
for(int i = 0 ; i < n ; i++){
for(int j = 0 ; j < m ; j++){
cin >> a[i][j];
}
}
cout<<"Elements in the wave form are: "<<endl;
for(int j = 0 ; j < m ; j++){
```

4.Given a positive integer n, generate a n x n matrix filled with elements from 1 to n2 in spiral order.
Input 1: n = 3
Output 1: [[1,2,3],[8,9,4],[7,6,5]]
Input 2: n = 1

Output 2: [[1]]
Solution :

```
#include<iostream>
using namespace std;
int main(){
int n ;
cout << "Enter the number of rows : ";
cin >> n;
int arr[n][n];
int k = 1, i = 0;
while( k <= n * n ){
int j = i;
// four steps
while( j < n - i ) // 1. horizontal, left to right
arr[i][j++] = k++;
j = i + 1;
while( j < n - i ) // 2. vertical, top to bottom
arr[j++][n-i-1] = k++;
j = n - i - 2;
while( j > i ) // 3. horizontal, right to left
arr[n-i-1][j--] = k++;
j = n - i - 1;
while( j > i ) // 4. vertical, bottom to top
arr[j--][i] = k++;
```

Q5. Predict the output :

```
int main(){
int a[][2] = {{1,2},{3,4}};
int i, j;
for (i = 0; i < 2; i++)
for (j = 0; j < 2; j++)
cout << a[i][j];
return 0;
}
```

Output :
1234