Algorithm Task

10 - Diagonal Difference

	ID	Full Name
1	20210159	إسلام مجدي محمد الصفي علي الجزار
2	20210052	أحمد رفعت كامل أحمد عفيفي
3	20211104	يوسف محمد نادي حسن مرسي
4	20210081	أحمد علي إبراهيم سليمان حرب
5	20210057	أحمد سعدالدين محمد علي المراكبي
6	20210050	أحمد رضا شعبان عبدالوهاب

1- Non-recursive:

1.1- Implementation:

```c
#include <stdio.h>
#include <math.h>
int diagonalDifference(int n, int arr[][n]) {
    int primaryDiagonal = 0, secondaryDiagonal = 0;
    for (int i = 0; i ⩽ n - 1; i++) {
        primaryDiagonal = primaryDiagonal + arr[i][i];
        secondaryDiagonal = secondaryDiagonal + arr[i][n - i - 1];
    }
    return abs(primaryDiagonal - secondaryDiagonal);
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n][n];
    for (int i = 0; i ⩽ n - 1; i++) {
        for (int j = 0; j ⩽ n - 1; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
    printf("%d", diagonalDifference(n, arr));
}
```

1.2- Documentation:

```
ALGORITHM DiagonalDifference(n, arr) {
        primaryDiagonal <- 0
        secondaryDiagonal <- 0
        for i <- 0 to n – 1 do
                primaryDiagonal <- primaryDiagonal + arr[i][i]
                secondaryDiagonal <- secondaryDiagonal + arr[i][n - i - 1]
        return abs(primaryDiagonal - secondaryDiagonal)
}
```

$$\sum_{i=0}^{n-1} 1 = n - 1 - 0 + 1 = n$$

So, Time Complexity is Θ(n)

```
3
11 2 4
4 5 6
10 8 -12
15
Process returned 0 (0x0)   execution time : 19.548 s
Press any key to continue.
```

2- Recursive:

2.1- Implementation:

```c
#include <stdio.h>
#include <math.h>
int diagonalDifference(int n, int arr[][n], int i) {
    if (i == n) {
        return 0;
    }
    return arr[i][i] - arr[i][n - i - 1] + diagonalDifference(n, arr, i + 1);
}
int main() {
    int n;
    scanf("%d", &n);
    int arr[n][n];
    for (int i = 0; i <= n - 1; i++) {
        for (int j = 0; j <= n - 1; j++) {
            scanf("%d", &arr[i][j]);
        }
    }
    printf("%d", abs(diagonalDifference(n, arr, 0)));
}
```

2.2- Documentation:

ALGORITHM DiagonalDifference(n, arr, i) {

If (i = n)

return 0

return arr[i][i] – arr[n – i – 1] + DiagonalDifference(n, arr, i + 1)

}

$$
\begin{aligned}
T(n) &= T(n-1) + 1 \\
T(n-1) &= T(n-2) + 1 \\
T(n) &= (T(n-2) + 1) + 1 \\
&= T(n-2) + 2 \\
T(n-2) &= T(n-3) + 1 \\
T(n) &= (T(n-3) + 1) + 2 \\
&= T(n-3) + 3 \\
T(n) &= T(n-k) + k \\
n - k = 1 &\implies k = n - 1 \\
T(n) &= T(1) + n - 1 \\
&= 1 + n - 1 \\
&= n
\end{aligned}
$$

So, Time Complexity is Θ(n)

```
3
11 2 4
4 5 6
10 8 -12
15
Process returned 0 (0x0)   execution time : 18.108 s
Press any key to continue.
```

3- Comparison

ALGORITHM	Time Complexity		
	Best Case	Average Case	Worst Case
Non-recursive	Ω(n)	Θ(n)	O(n)
Recursive	Ω(n)	Θ(n)	O(n)

Both have the same Time Complexity, but Non-recursive is better than Recursive