

1. Make 2D Array in C++ and print left diagonal and right diagonal sum of a 3x3 matrix.

```
#include<iostream>
using namespace std;

int main() {
    int array[3][3], leftdi=0, rightdi=0;
    cout<<"enter values from left to right then proceeding downwards"<<endl;
    for(int i=0; i<=2; i++){
        for (int j=0; j<=2; j++){
            cout<<"enter value:";
            cin>>array[i][j];
        }
    }

    for (int i=0; i<=2; i++){
        leftdi+=array[i][i];
    }

    cout<<"sum of left diagonal is "<<leftdi<<endl;
    for (int i=0; i<=2; i++){
        rightdi+=array[i][2-i];
    }
    cout<<"sum of right diagonal is "<<rightdi;
}
```

```
enter values from left to right then proceeding downwards
enter value:1
enter value:2
enter value:3
enter value:4
enter value:5
enter value:6
enter value:7
enter value:8
enter value:9
sum of left diagonal is 15
sum of right diagonal is 15
Process returned 0 (0x0)   execution time : 6.466 s
Press any key to continue.
```

2. Write a function to add two 2D arrays of size 3x3.

```

void addarrays(int array1[3][3],int array2[3][3],int array3[3][3]){
    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            array3[i][j]=array1[i][j]+array2[i][j];
        }
    }
}

int main()
{
    int array2[3][3],array1[3][3];
    int array3[3][3]={0,0,0},{0,0,0},{0,0,0};
    cout <<"ENTER VALUES FOR ARRAY1"<< endl;
    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            cin>>array1[i][j];
        }
    }
    cout <<"ENTER VALUES FOR ARRAY2"<< endl;
    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            cin>>array2[i][j];
        }
    }

    addarrays(array1,array2,array3);
    cout <<"SUM OF ARRAYS IS"<< endl;

    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            cout <<array3[i][j]<< endl;
        }
    }

    return 0;
}

```

ENTER VALUES FOR ARRAY1

1

2

3

4

5

6

7

8

9

ENTER VALUES FOR ARRAY2

1

2

3

4

5

6

7

8

9

SUM OF ARRAYS IS

2

4

6

8

10

12

14

16

18

3. Using 2D arrays in C++, take transpose of a 3x3 matrix. Make a transpose function

```
const int rows=3,col=3;

void transpose(int array[rows][col]) {
    int temp;
    for (int i= 0; i < col; ++i) {
        for (int j = i+1; j < col; j++) {
            temp = array[i][j];
            array[i][j] = array[j][i];
            array[j][i] = temp;
        }
    }
}

int main()
{
    int array1[3][3];
    cout <<"Enter your values for first array"<< endl;
    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            cin>>array1 [i][j];
        }
    }
    transpose(array1);
    cout <<"transpose of the array is"<< endl;
    for(int i=0;i<=2;i++){
        for(int j=0;j<=2;j++){
            cout<<array1[i][j];
        }
    }
    cout<<endl;
    return 0;
}
```

Enter your values for first array

1
2
3
4
5
6
7
8
9
transpose of the array is
147
258
369

Process returned 0 (0x0) execution time : 4.598 s
Press any key to continue.

4.Using 2D arrays in C++, implement 3x3 matrix multiplication.

Make a function.

```
void multiplication(int array1[][col], int array2[][col], int result[][col]) {
    for (int i=0; i < col; i++) {
        for (int j = 0; j < col; j++) {
            result[i][j] = 0;
            for (int z = 0; z < col; z++) {
                result[i][j] += array1[i][z] * array2[z][j];
            }
        }
    }
}

int main()
{
    int array2[3][3], array1[3][3];
    int array3[3][3]={{0,0,0},{0,0,0},{0,0,0}};
    cout << "Enter your values for first array" << endl;
    for (int i=0; i<=2; i++) {
        for (int j=0; j<=2; j++) {
            cin >> array1[i][j];
        }
        cout << "Enter your values for second array" << endl;
        for (int i=0; i<=2; i++) {
            for (int j=0; j<=2; j++) {
                cin >> array2[i][j];
            }
        }
        cout << "your resultant array is" << endl;
        multiplication(array1, array2, array3);
        cout << "product of both arrays is" << endl;

        for (int i=0; i<=2; i++) {
            for (int j=0; j<=2; j++) {
                cout << array3[i][j] << endl;
            }
        }
    }
}
```

```
Enter your values for first array
1
2
3
4
5
6
7
8
9
Enter your values for second array
1
2
3
4
5
6
7
8
9
your resultant array is
addition of both arrays is
30
36
42
66
81
96
102
126
```

5.Print the multiplication table of 15 using recursion.

```
#include <iostream>

using namespace std;

void multiplationtable(int mynum, int start) {
    if (start > 10) {
        return;
    }
    else{
        int product=mynum * start;
        cout<<mynum<<"*"<<start<<"="<<product<<endl;
        multiplationtable(mynum,start+1);
    }
}

int main(){
    int start=0;
    cout << "table of 15 is" << endl;
    multiplationtable(15,start);
    return 0;
}
```

```
table of 15 is
15*0=0
15*1=15
15*2=30
15*3=45
15*4=60
15*5=75
15*6=90
15*7=105
15*8=120
15*9=135
15*10=150
```

```
Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```

