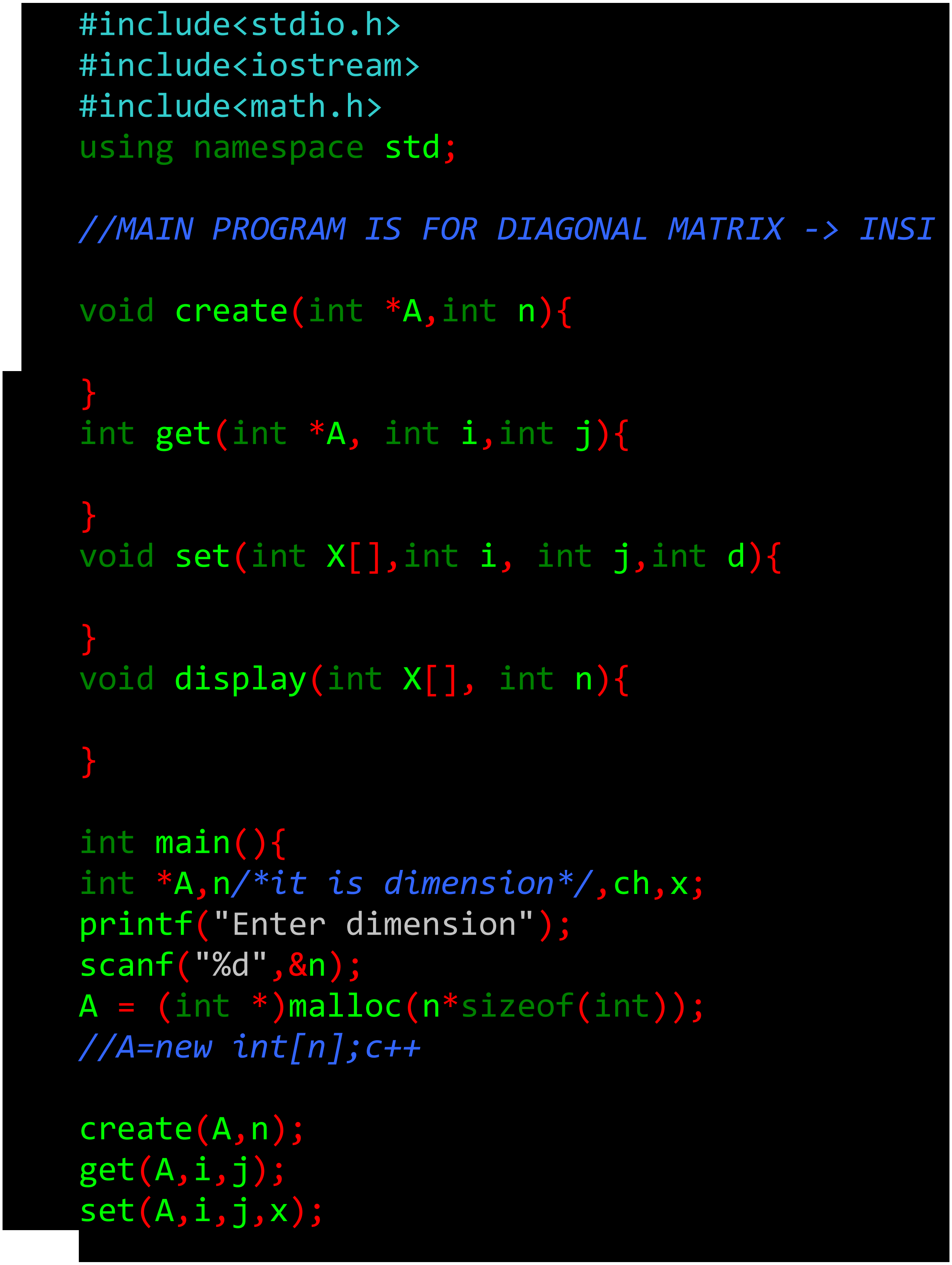
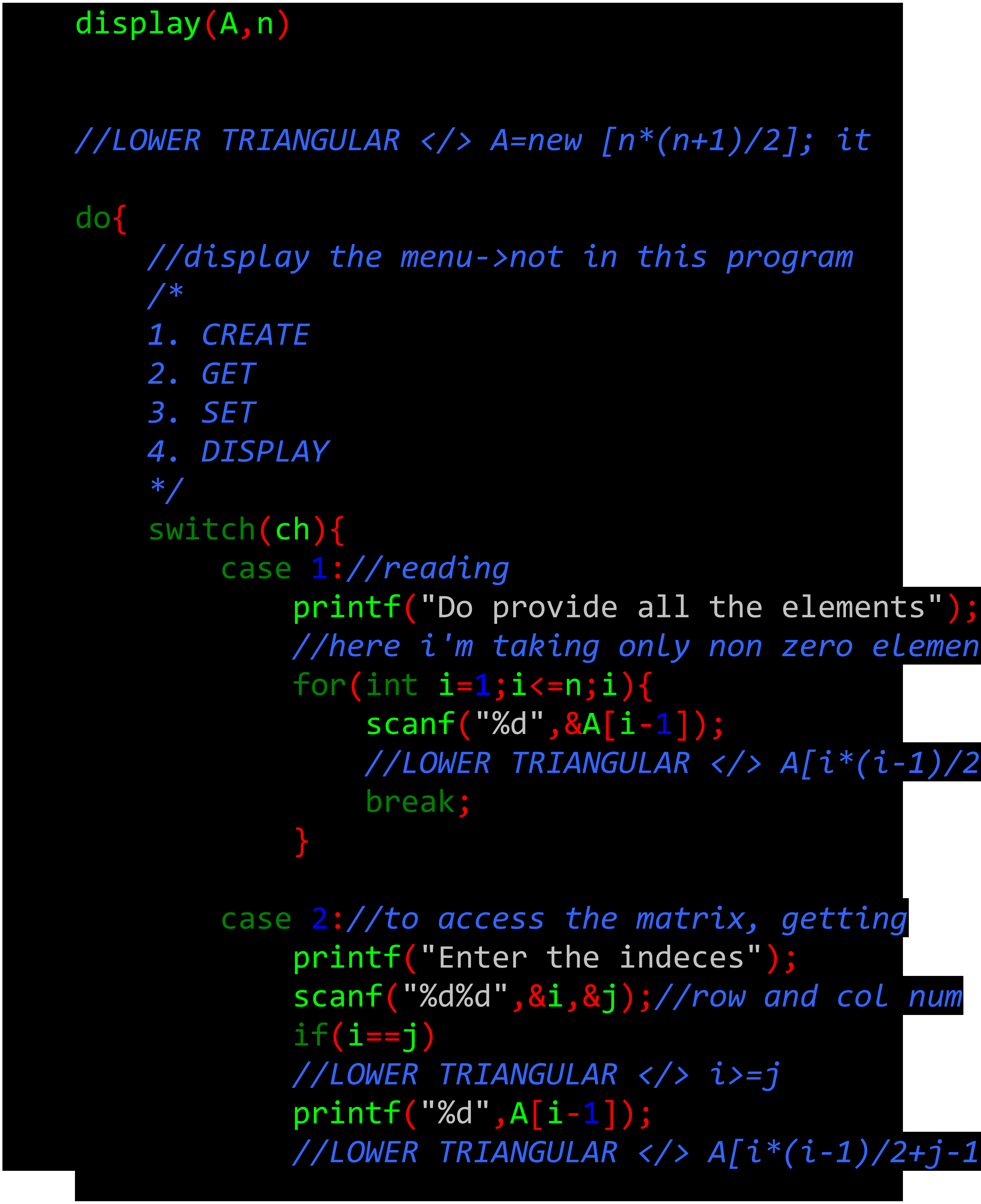
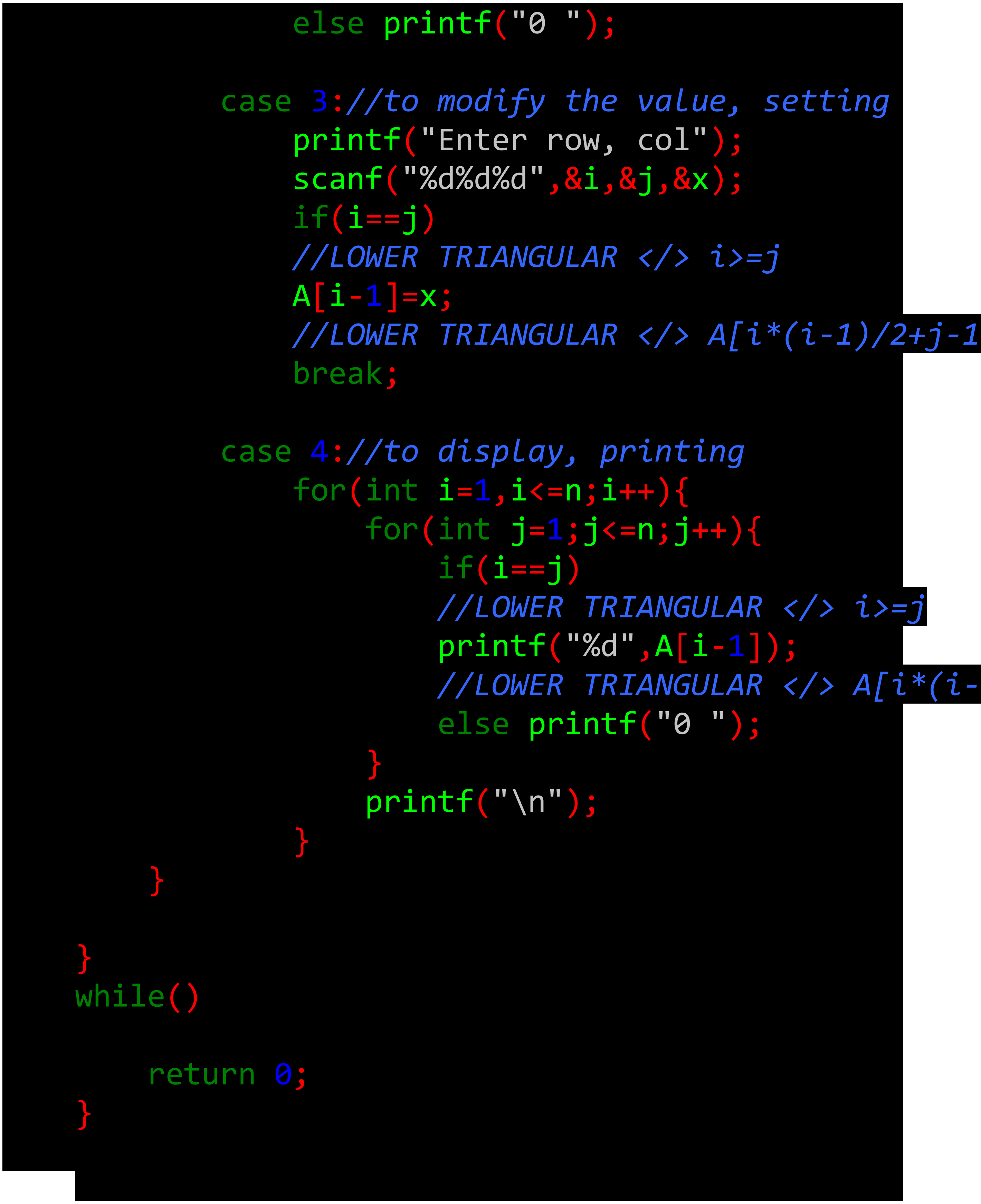


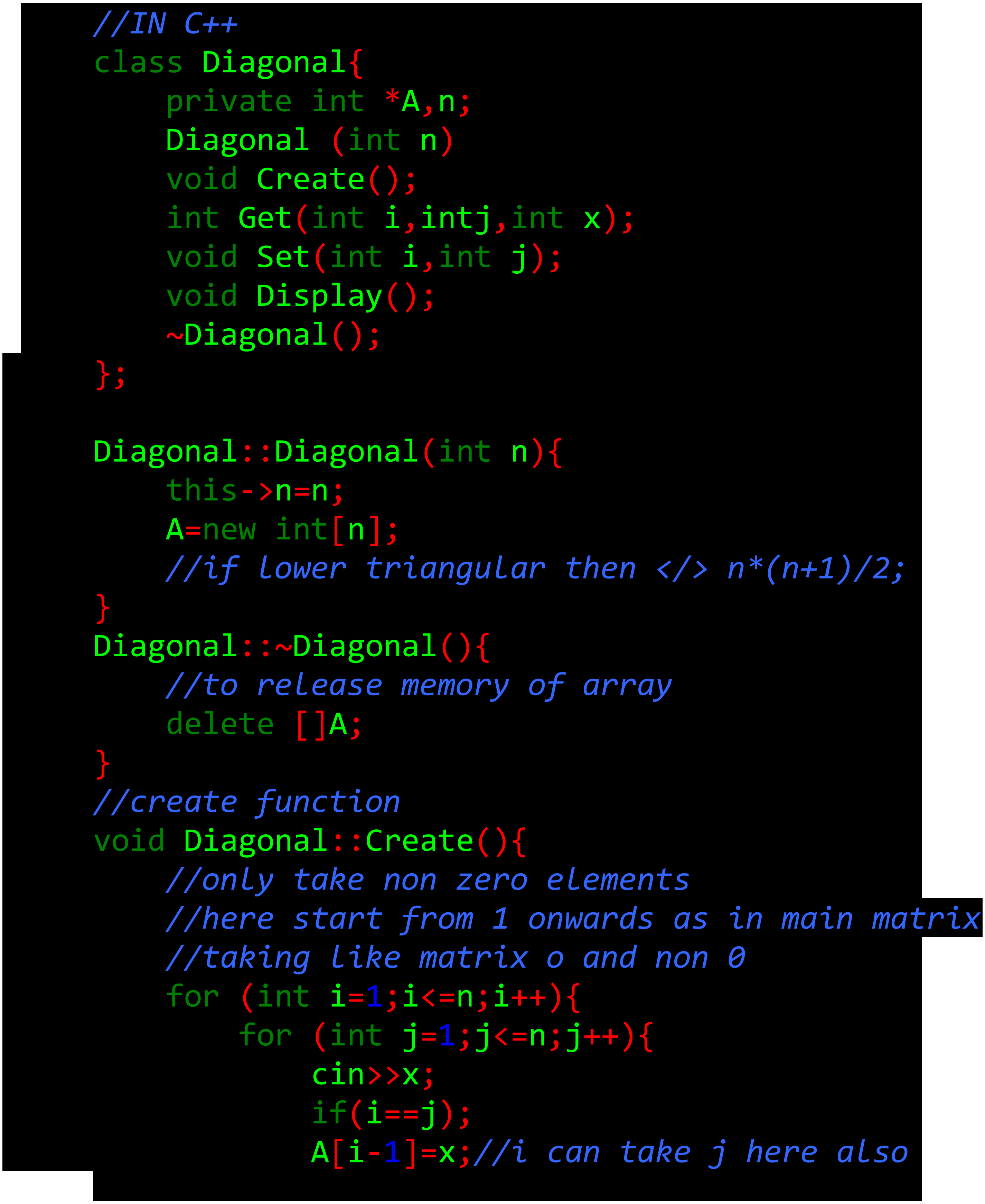
```
#include<stdio.h>
#include<iostream>
#include<math.h>
using namespace std;

//MAIN PROGRAM IS FOR DIAGONAL MATRIX -> INSI

void create(int *A,int n){

}
int get(int *A, int i,int j){

}
void set(int X[],int i, int j,int d){

}
void display(int X[], int n){

}

int main(){
int *A,n/*it is dimension*/,ch,x;
printf("Enter dimension");
scanf("%d",&n);
A = (int *)malloc(n*sizeof(int));
//A=new int[n];c++

create(A,n);
get(A,i,j);
set(A,i,j,x);
```



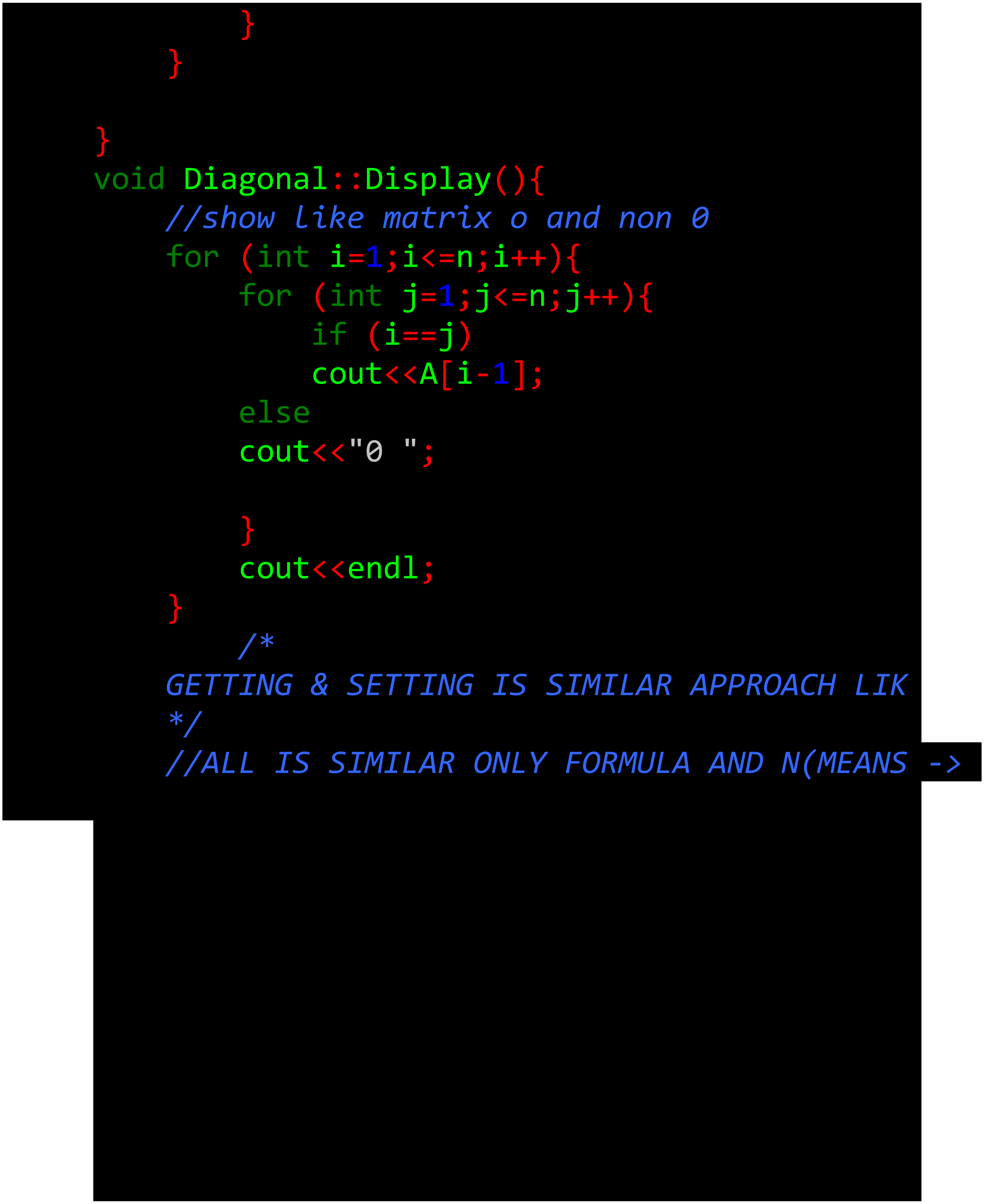
```
    display(A,n)


    //LOWER TRIANGULAR </> A=new [n*(n+1)/2]; it

    do{
        //display the menu->not in this program
        /*
        1. CREATE
        2. GET
        3. SET
        4. DISPLAY
        */
        switch(ch){
            case 1://reading
                printf("Do provide all the elements");
                //here i'm taking only non zero elemen
                for(int i=1;i<=n;i){
                    scanf("%d",&A[i-1]);
                    //LOWER TRIANGULAR </> A[i*(i-1)/2
                    break;
                }

            case 2://to access the matrix, getting
                printf("Enter the indeces");
                scanf("%d%d",&i,&j);//row and col num
                if(i==j)
                //LOWER TRIANGULAR </> i>=j
                printf("%d",A[i-1]);
                //LOWER TRIANGULAR </> A[i*(i-1)/2+j-1
```



```
            else printf("0 ");

        case 3://to modify the value, setting
            printf("Enter row, col");
            scanf("%d%d%d",&i,&j,&x);
            if(i==j)
            //LOWER TRIANGULAR </> i>=j
            A[i-1]=x;
            //LOWER TRIANGULAR </> A[i*(i-1)/2+j-1
            break;

        case 4://to display, printing
            for(int i=1,i<=n;i++){
                for(int j=1;j<=n;j++){
                    if(i==j)
                    //LOWER TRIANGULAR </> i>=j
                    printf("%d",A[i-1]);
                    //LOWER TRIANGULAR </> A[i*(i-
                    else printf("0 ");
                }
                printf("\n");
            }
    }

}
while()

    return 0;
}
```



```
//IN C++
class Diagonal{
    private int *A,n;
    Diagonal (int n)
    void Create();
    int Get(int i,intj,int x);
    void Set(int i,int j);
    void Display();
    ~Diagonal();
};

Diagonal::Diagonal(int n){
    this->n=n;
    A=new int[n];
    //if lower triangular then </> n*(n+1)/2;
}
Diagonal::~Diagonal(){
    //to release memory of array
    delete []A;
}
//create function
void Diagonal::Create(){
    //only take non zero elements
    //here start from 1 onwards as in main matrix
    //taking like matrix o and non 0
    for (int i=1;i<=n;i++){
        for (int j=1;j<=n;j++){
            cin>>x;
            if(i==j);
            A[i-1]=x;//i can take j here also
```



```
            }
        }

    }
    void Diagonal::Display(){
        //show like matrix o and non 0
        for (int i=1;i<=n;i++){
            for (int j=1;j<=n;j++){
                if (i==j)
                cout<<A[i-1];
            else
            cout<<"0 ";

            }
            cout<<endl;
        }
            /*
        GETTING & SETTING IS SIMILAR APPROACH LIK
        */
        //ALL IS SIMILAR ONLY FORMULA AND N(MEANS ->
```