Diagonal Matrix CPP

```cpp
#include <iostream>

using namespace std;

class Diagonal
{
private:
    int *A;
    int n;
public:
    Diagonal()
    {
        n=2;
        A=new int[2];
    }
    Diagonal(int n)
    {
        this->n=n;
        A=new int[n];
    }
    ~Diagonal()
    {
        delete []A;
    }
    void Set(int i,int j,int x);
    int Get(int i,int j);
    void Display();
    int GetDimension(){return n;}
};

void Diagonal::Set(int i,int j,int x)
{
    if(i==j)
        A[i-1]=x;
}

int Diagonal::Get(int i,int j)
{
    if(i==j)
        return A[i-1];
    return 0;
}
void Diagonal::Display()
{
```

```
    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;
    }
}

int main()
{
    int d;
    cout<<"Enter Dimensions";
    cin>>d;

    Diagonal dm(d);

    int x;
    cout<<"Enter All Elements";
    for(int i=1;i<=d;i++)
    {
        for(int j=1;j<=d;j++)
        {
            cin>>x;
            dm.Set(i,j,x);
        }
    }

    dm.Display();

    return 0;
}
```