#include <iostream>

using namespace std;

class Recurrence

{

public:

*// T(n) = T(n - 1) + n , T(0) = 1*

    int relation\_1(int n)

    {

        if (n == 0)

            return 1;

        return relation\_1(n - 1) + n;

    }

*// T(n) = T(n - 1) + n^2 , T(0) = 1*

    int relation\_2(int n)

    {

        if (n == 0)

            return 1;

        return relation\_2(n - 1) + n \* n;

    }

*// T(n) = 2T(n/2) + n , T(1) = 1 , T(0) = 0*

    int relation\_3(int n)

    {

        if (n <= 1)

            return n;

        return 2 \* relation\_3(n / 2) + n;

    }

};

int main()

{

    int n;

    Recurrence R;

    cout << "Enter an integer : ";

    cin >> n;

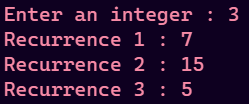
    cout << "Recurrence 1 : " << R.relation\_1(n) << endl;

    cout << "Recurrence 2 : " << R.relation\_2(n) << endl;

    cout << "Recurrence 3 : " << R.relation\_3(n) << endl;

    return 0;

}



Q15)

#include <iostream>

#include <cmath>

using namespace std;

class Polynomial

{

    int degree;

    int \*coefficients;

public:

    Polynomial(int d)

    {

        degree = d;

        coefficients = new int[d + 1];

    }

    void inputCoefficients()

    {

        for (int d = degree; d >= 0; d--)

            cin >> coefficients[d];

    }

    void display()

    {

        for (int d = degree; d > 0; d--)

            cout << coefficients[d] << "x^" << d << " + ";

        cout << coefficients[0] << "x^0" << endl;

    }

    int f(int n)

    {

        int res = 0;

        for (int d = degree; d >= 0; d--)

            res += coefficients[d] \* pow(n, d);

        return res;

    }

};

int main()

{

    int n,m;

    cout << "Enter degree of polynomial : ";

    cin >> n;

    Polynomial p(n);

    cout << "Enter coefficients of polynomial" << endl;

    p.inputCoefficients();

    cout << "Polynomial : ";

    p.display();

    cout << "Enter a number : ";

    cin >> m;

    cout << "Value at " << m << " : " << p.f(m);

    return 0;

}

