

Rta 1:

```
#include <iostream>
```

```
using namespace std;
```

```
class Numeros {
```

```
public:
```

```
    int cargarValor() {
```

```
        int valor;
```

```
        cout << "Ingrese un valor entero: ";
```

```
        cin >> valor;
```

```
        return valor;
```

```
    }
```

```
    void mostrarNumerosHasta(int limite) {
```

```
        if (limite <= 0) {
```

```
            cout << "El valor ingresado debe ser mayor que cero." << endl;
```

```
            return;
```

```
        }
```

```
        cout << "Números desde 1 hasta " << limite << ":" << endl;
```

```
        for (int i = 1; i <= limite; ++i) {
```

```
            cout << i << " ";
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
};
```

```
int main() {
```

```
    Numeros numeros;
```

```
    int valorIngresado = numeros.cargarValor();
```

```
    numeros.mostrarNumerosHasta(valorIngresado);
```

```
    return 0;
```

```
}
```

Rta 2:

```
#include <iostream>
```

```
using namespace std;
```

```
class Comparador {
```

```
private:
```

```
    int obtenerMayor(int num1, int num2) {
```

```
        return (num1 > num2) ? num1 : num2;
```

```
    }
```

```
public:
```

```

void cargarEnteros() {
    int num1, num2;

    cout << "Ingrese el primer entero: ";
    cin >> num1;
    cout << "Ingrese el segundo entero: ";
    cin >> num2;

    int mayor = obtenerMayor(num1, num2);
    cout << "El mayor de los dos enteros es: " << mayor << endl;
}
};

```

```

int main() {
    Comparador comparador;
    comparador.cargarEnteros();

    return 0;
}

```

Rta 3:

```

#include <iostream>
using namespace std;

```

```

class CNumeros {
private:
    int n1, n2, n3;

```

```

public:

```

```

    CNumeros(int num1, int num2, int num3) {
        n1 = num1;
        n2 = num2;
        n3 = num3;
    }

```

```

    int calcularMaximo() {
        int maximo = n1;
        if (n2 > maximo) {
            maximo = n2;
        }
        if (n3 > maximo) {
            maximo = n3;
        }
        return maximo;
    }
}

```

```

int calcularMinimo() {
    int minimo = n1;
    if (n2 < minimo) {
        minimo = n2;
    }
    if (n3 < minimo) {
        minimo = n3;
    }
    return minimo;
}
};

```

```

int main() {
    int num1, num2, num3;

    cout << "Ingrese el primer número: ";
    cin >> num1;
    cout << "Ingrese el segundo número: ";
    cin >> num2;
    cout << "Ingrese el tercer número: ";
    cin >> num3;

    CNumeros numeros(num1, num2, num3);

    int maximo = numeros.calcularMaximo();
    int minimo = numeros.calcularMinimo();

    cout << "El máximo de los tres números es: " << maximo << endl;
    cout << "El mínimo de los tres números es: " << minimo << endl;

    return 0;
}

```

Rta 4:

```

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

class CEmpleados {
private:
    string nombre;
    float salario;
    int horasTrabajoPorDia;

public:

    CEmpleados(string nom, float sal, int horas) {

```

```

        nombre = nom;
        salario = sal;
        horasTrabajoPorDia = horas;
    }

    void aumentarSalario() {
        if (horasTrabajoPorDia == 8) {
            salario += 250000;
        } else if (horasTrabajoPorDia == 10) {
            salario += 300000;
        } else if (horasTrabajoPorDia >= 12) {
            salario += 350000;
        }
    }

    void imprimirDatos() {
        cout << "Nombre: " << nombre << endl;
        cout << "Salario final: $" << salario << endl;
    }
};

int main() {
    string nombre;
    float salario;
    int horasTrabajoPorDia;

    cout << "Ingrese el nombre del empleado: ";
    getline(cin, nombre);
    cout << "Ingrese el salario del empleado: ";
    cin >> salario;
    cout << "Ingrese las horas de trabajo por día del empleado: ";
    cin >> horasTrabajoPorDia;

    CEmpleados empleado(nombre, salario, horasTrabajoPorDia);

    empleado.aumentarSalario();

    empleado.imprimirDatos();

    return 0;
}

```

Rta 5:

```

#include <iostream>
#include <vector>

using namespace std;

class CNumeros {
public:

    vector<int> obtenerNumerosPerfectos(int N) {
        vector<int> perfectos;
        for (int i = 1; i <= N; ++i) {
            if (esNumeroPerfecto(i)) {
                perfectos.push_back(i);
            }
        }
        return perfectos;
    }

private:

    bool esNumeroPerfecto(int num) {
        if (num <= 1) {
            return false;
        }
        int sumaDivisores = 1;
        for (int i = 2; i * i <= num; ++i) {
            if (num % i == 0) {
                sumaDivisores += i;
                if (i != num / i) {
                    sumaDivisores += num / i;
                }
            }
        }
        return sumaDivisores == num;
    }
};

int main() {
    int N;
    cin >> N;

    CNumeros numeros;
    vector<int> perfectos = numeros.obtenerNumerosPerfectos(N);

    for (int num : perfectos) {
        cout << num << endl;
    }
}

```

```
    return 0;  
}
```