

# SUMATORIAS

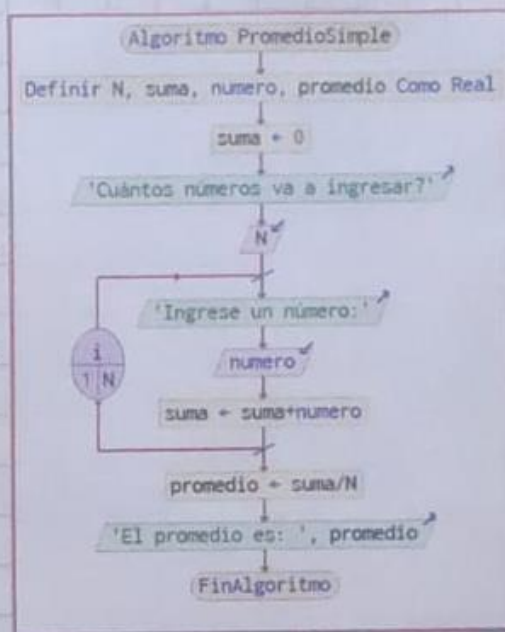
- ① Realizar un programa que permita visualizar el promedio de  $N$  números ingresados por teclado..

## Análisis

**Entrada:** Se pide al usuario la cantidad de dígitos que va a ingresar

**Proceso:** El programa lee esos números los suma y luego calcula el promedio, dividiendo la suma entre la cantidad de Números

**Salida:** El promedio se muestra al usuario



## CODIGO

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

int main () {
    int N;
    double suma = 0, numero;

    cout << "Cuántos números va a ingresar";
    cin >> N;

    for (int i=0; i<N; i++) {
        cout << "Ingrese un número: ";
        cin >> numero;
        suma += numero;
    }

    cout << "El promedio es: " << suma / N << endl;
    return 0;
}
```

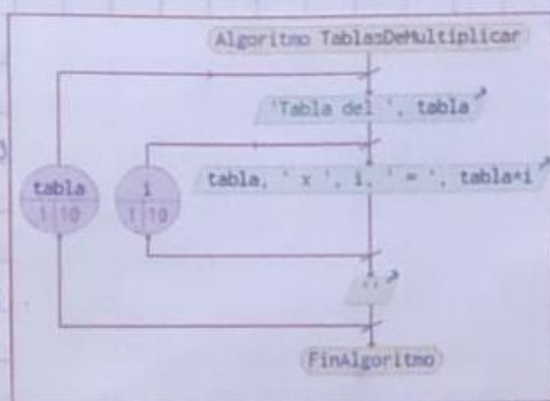
2) Realizar un programa que permita visualizar las tablas de Multiplicar  
Tablas de Multiplicar 1-10

### ANALISIS

Entrada: No se requiere entrada del usuario para generar las tablas

Proceso: El programa genera las tablas de multiplicar del 1-10 usando un bucle anidado: un bucle externo para el numero de la tabla (1a10) y uno interno para las multiplicaciones (1a10)

Salida: Muestra las tablas del 1 al 10 en la consola



### CODIGO

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

```
int main () {
```

```
    for (int tabla = 1; tabla <= 10; tabla++) {
```

```
        cout << "tabla del " << tabla << endl;
```

```
        for (int i = 1; i <= 10; i++) {
```

```
            cout << tabla << " x " << i << " = " << tabla * i << endl;
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
    return 0;
```

```
}
```



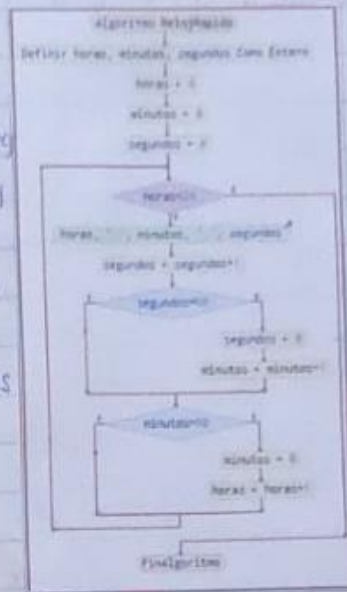
13) Realizar un programa que permita simular un "reloj rápido" que muestre en la pantalla las horas, minutos y segundos.  
RELOJ

### ANALISIS

**Entrada:** No se necesita entrada del usuario. El Programa incrementara automaticamente seg.

**Proceso:** El programa debe incrementar los segundos cada vez que llegue a 60 debe incrementar los minutos. Si los minutos llegan a 60 se incrementara las horas. El reloj simula que el avance rapidamente.

**Salida:** Muestra las horas, minutos y segundos actualizados en la pantalla en un ciclo.



### CODIGO

```

#include <iostream>
#include <thread>
#include <chrono>
using namespace std;

int main () {
    int horas = 0, minutos = 0, segundos = 0;
    while (horas < 24) {
        cout << (horas < 10 ? "0" : "") << horas << " ";
        << (minutos < 10 ? "0" : "") << minutos << " ";
        << (segundos < 10 ? "0" : "") << segundos << endl;
        This_thread::sleep_for (chrono::milliseconds(100)); // Segundos++;
        if (segundos == 60) {
            segundos = 0;
            minutos++;
        }
        if (minutos == 60) {
            minutos = 0;
            horas++;
        }
    }
    return 0;
}
  
```

4) Realizar un programa que permita leer  $N$  números e indique el Mayor

### ANÁLISIS

**Entrada:** El usuario debe ingresar cuantos números ( $N$ ) va a proporcionar

**Proceso:** El programa lee los  $N$  números y determina cuál es el Mayor

**Salida:** Muestra el número más grande entre los ingresados

### CODIGO

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

```
int main () {
    int N, numero mayor = -999999;
```

```
    cout << "Cuantos numeros va a ingresar ? ";
    cin >> N;
```

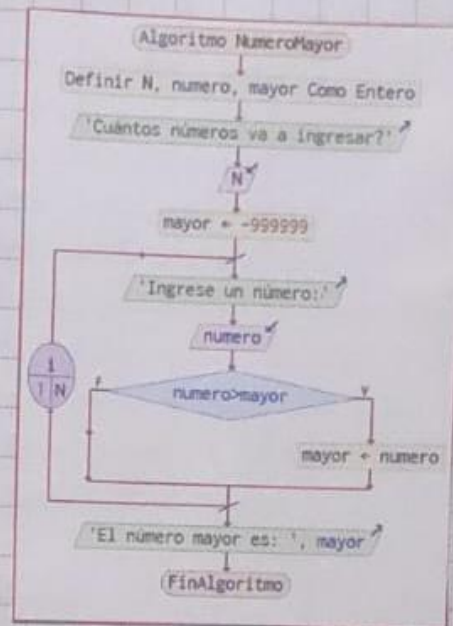
```
    for (int i = 0; i < N; i++) {
        cout << "ingrese un numero ";
        cin >> numero;
        if (numero > mayor) {
            mayor = numero;
        }
    }
```

```
}
```

```
    cout << "El numero mayor es: " << Mayor << endl;
```

```
    return 0;
```

```
}
```





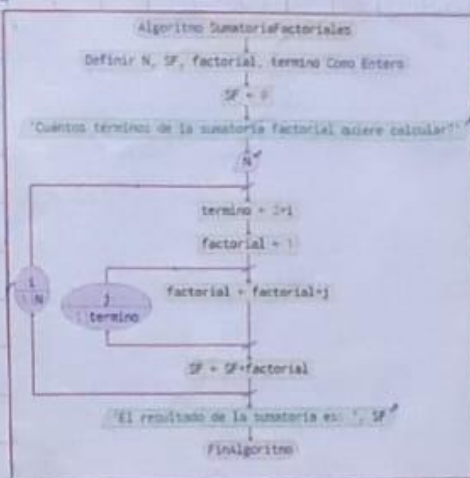
- 5) Evaluar la siguiente sumatoria de  $N$  términos almacenados el resultado en `sum`. Desplegar el resultado  $2! + 4! + 6! + \dots$

### ANÁLISIS

**Entrada** "El usuario debe ingresar el número de términos  $NNN$  que determinarán cuántos números pares se utilizaron para calcular sus factoriales"

**Proceso** "El programa calcula el factorial de los números pares empezando desde 2 hasta el término  $NNN$  y suma esas factoriales"

**Salida** se muestra la sumatoria total de los factoriales almacenando la variable



### CODIGO

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

```
int main () {
    int N; termino, factorial;
    long long SF = 0;
```

```
    cout << "Cuántos terminos de la sumatoria factorial quiere calcular?"
    cin >> N;
```

```
    for (int i = 1; i <= N; i++) {
        termino = 2 * i;
        factorial = 1;
        for (int j = 1; j <= termino; j++) {
            factorial *= j;
        }
        SF += factorial;
    }
```

```
    cout << "el resultado de la sumatoria es:" << SF << endl;
```

```
    return 0;
```

```
}
```

6) Evaluar la siguiente sumatoria de N terminos almacenados el resultado en ST. Desplegar el resultado.

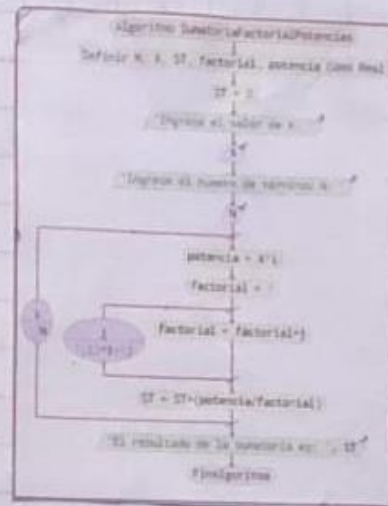
$$ST = \frac{x}{1!} + \frac{x^2}{3!} + \frac{x^3}{5!} + \frac{x^4}{7!} + \dots$$

### ANALISIS

**Entrada:** El usuario ingresa dos valores NNN (el numero de terminos de la serie) y xxx (la base que se eleva a las distintas potencias)

**Proceso:** El programa calcula la serie usando el numero de terminos NNN y la base xxx donde cada termino el exponent de xxx y el factorial son impares.

**Salida:** El resultado de la sumatoria es almacenado en la variable ST y luego Mostrado



### CODIGO

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

```
int main () {
```

```
    int N;
```

```
    double x, ST=0, potencia, factorial;
```

```
    cout << "ingrese el valor de x: ";
    cin >> x;
```

```
    cout << "ingrese el numero de terminos N: ";
    cin >> N;
```

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

```
        potencia = 1;
```

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

```
            potencia *= x;
```

```
        factorial = 1;
```

```
        for (int j=1; j<= (2*i-1); j++) {
```

```
            factorial *= j;
```

```
        }
```

```
        ST += potencia / factorial;
```

```
    }
```

```
    cout << "el resultado de la sumatoria es: " << ST << endl;
```

```
    return 0;
```

```
}
```



#) Evaluar la siguiente sumatoria de N términos almacenados el resultado en ST. Desplegar resultado.

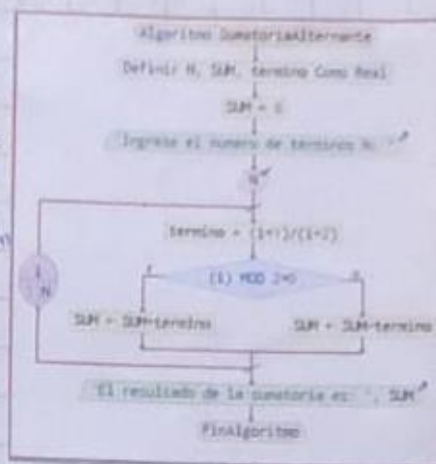
$$SUM = \frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \frac{5}{4} + \dots + \frac{N}{N-1}$$

### ANÁLISIS

**Entrada:** El usuario ingrese el valor de NNN que determina cuantos términos se sumaran.

**Desarrollo:** El programa calcula la sumatoria alternante con NNN términos donde el numerador comienza en 2 y el denominador en 3 incrementando en cada término y alternando en la suma y resta.

**Salida:** El resultado de la sumatoria es almacenada en la variable SUM y luego mostrada.



### CODIGO

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

```
int main () {
    int N;
    double SUM=0; término;
```

```
    cout<< "Ingrese el numero de términos N: ";
    cin>> N;
```

```
    for (int i = 1; i <= N; i++) {
        término = (double)(i+1)/(i+2);
        if (i%2 == 0) {
            SUM -= término;
        } else {
            SUM += término;
        }
    }
```

```
    cout<< "El resultado de la sumatoria es: " << SUM << endl;
```

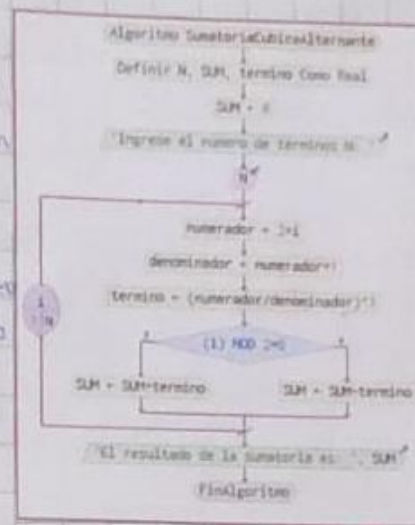
```
    return 0;
}
```

8) Evaluar la siguiente sumatoria de  $N$  términos almacenados el resultado en SUM. Desplegar Resultado

$$SUM = \frac{2}{3^3} + \frac{4}{5^3} + \frac{6}{7^3} + \frac{8}{9^3} + \dots$$

## ANÁLISIS

**Entrada:** El usuario ingresa el valor de  $N$ , que determina cuántos términos suman.  
**Proceso:** el programa calcula alternadamente la sumatoria de  $N$  donde los numeradores son pares y los denominadores también siguen un patrón y cada tracción se eleva al cubo.  
**Salida:** El resultado de la sumatoria almacenado en la variable SUM y luego Mostrado



## CODIGO

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

```
int main () {
    int N;
    double SUM = 0, término;
```

```
    cout << "Ingrese el número de términos N: ";
    cin >> N;
```

```
    for (int i = 1; i <= N; i++) {
        double numerador = 2 * i;
        double denominador = numerador + 1;
        término = pow(numerador / denominador, 3);
```

```
        if (i % 2 == 0) {
            SUM += término;
        } else {
            SUM -= término;
        }
    }
```

```
    cout << "El resultado de la sumatoria es " << SUM << endl;
```

```
    return 0;
```

```
}
```



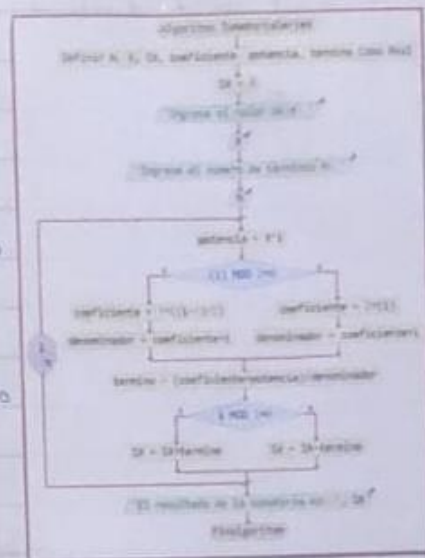
9) Evaluar la siguiente sumatoria de  $N$  terminos almacenados en  $Sx$   
 Desplegar Resultado  $Sx \quad \frac{x}{1} - \frac{x^2}{2} + \frac{2x^3}{6} - \frac{2x^4}{8} + \frac{3x^5}{15} - \frac{3x^6}{18} + \dots$

### ANALISIS

Entrada El usuario ingresa el valor  $NNN$  (numero de terminos) y  $xxx$ .

Proceso: El programa calcula la sumatoria alternante de  $NNN$  terminos con la formula dada teniendo en cuenta que los coeficientes y las potencias de  $xxx$  cambia en cada termino

Salida El resultado de la sumatoria es almacenada en variable  $Sx$  y luego mostrado



### CODIGO

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

```
int main () {
    int N;
    double x, Sx = 0, c, p, t;
```

```
    cout << "Ingrese el valor de x : ";
    cin >> x;
    cout << "Ingrese el numero de terminos N: ";
    cin >> N;
```

```
    for (int i = 1; i <= N; i++) {
        p = pow(x, i);
        if (i % 2 == 0) {
            c = 1 * (i / 2);
            t = (c * p) / (c * i);
            Sx -= t;
        } else {
            c = 1 * (i - 1) / 2;
            t = (c * p) / (c * i);
            Sx += t;
        }
    }
```

```
    cout << "el resultado de la sumatoria es " << Sx << endl;
    return 0;
}
```

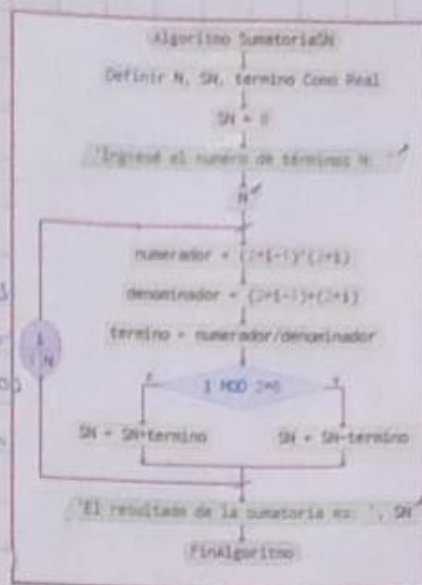
10) Evaluar la siguiente sumatoria de  $N$  términos almacenados el resultado en  $SN$ .  
Desplegar resultado  $SN = \frac{1^2}{1 \times 2} - \frac{3^4}{3 \times 4} + \frac{5^6}{5 \times 6} - \frac{7^8}{7 \times 8} + \dots$

### ANÁLISIS

**Entrada** El usuario ingresa el valor de  $NNN$  que determina cuántos términos de la sumatoria se calcularán.

**Desarrollo** El programa calcula la sumatoria alternante de  $NNN$  términos donde los numeradores son potencias de números impares y los denominadores son productos de números consecutivos.

**Salida** El resultado de la sumatoria es almacenado en la memoria y luego mostrado.



### CODIGO

```

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

int main() {
    int
    double SN=0; numerador, denominador, termino;

    cout << "Ingrese el numero de terminos N: ";
    cin >> N;
    for (int i=1; i <= N; i++) {
        numerador = pow(2*i-1, 2*i);
        denominador = (2*i-1) * (2*i);

        if (i%2 == 0) {
            SN -= termino;
        } else {
            SN += termino;
        }
    }
    cout << "El resultado de la sumatoria es: " << SN << endl;
    return 0;
}
  
```



11) Evaluar la siguiente sumatoria de  $N$  términos almacenados el resultado en  $ST$  Desplegar el resultado

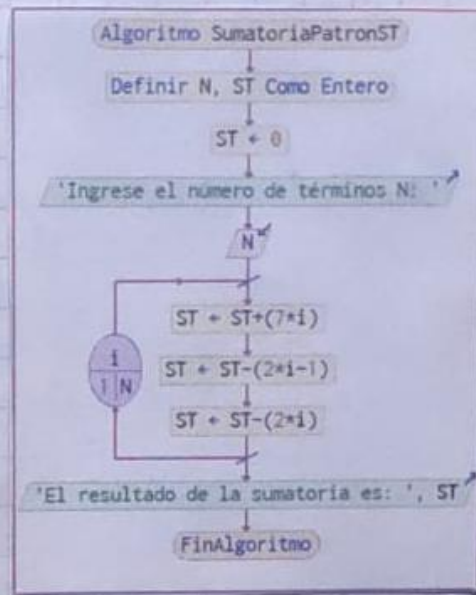
$$ST = 7 - 1 - 2 + 14 - 3 - 4 + 21 - 5 - 6 + \dots$$

### ANALISIS

Entrada: El usuario ingresa el valor de  $NNN$  que determina cuantos grupos de tres terminos se calcularan

Proceso: el programa sigue el patron: sumar multiples de 7 y restar dos numeros consecutivos despues de cada multiple

Salida: el resultado de la sumatoria es almacenado en la variable  $ST$  y mostrada



### CODIGO

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

```
int main () {
    int N, ST=0;
```

```
    cout << "Ingrese el numero de terminos N: ";
    cin >> N;
```

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

```
        ST += 7*i;
        ST -= (2*i - 1);
        ST -= 2*i;
```

```
    }
```

```
    cout << "El resultado de la sumatoria es" << ST << endl;
```

```
    return 0;
```

```
}
```

12) Evaluar la siguiente sumatoria de N términos almacenados el resultado en Sxy, x e y son datos de entrada; Desplegar Resuelto

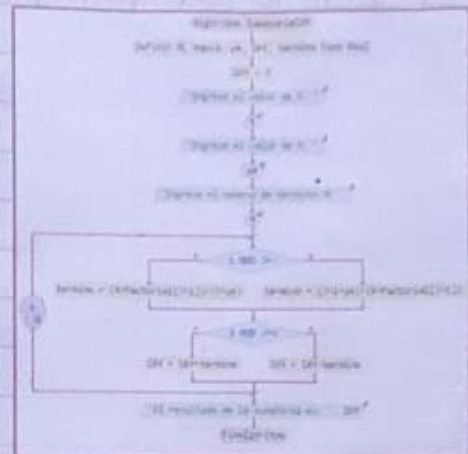
$$Sxy = \frac{2y}{x^3!} - \frac{x^6!}{5y} + \frac{8y}{x^9!} - \frac{x^{12!}}{11y} + \frac{14y}{x^{15!}} \dots$$

### ANÁLISIS

**Entrada:** El usuario ingresa los valores de xxx NNN y yyy que determinan cuantos terminos de la sumatoria se calcularian

**Proceso:** El programa sigue el patron de terminos que involucran factoriales xxx y xxx alternando entre sumas y restas

**Salida:** el resultado de la sumatoria es almacenada en la variable Sxy y luego mostrado



### CODIGO

```

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

long long factorial (int n){
    long long fact=1;
    for (int i=1; i<=n; i++){
        fact*=i;
    }
    return fact;
}

int main
{
    int N;
    double x, y, Sxy=0, termino;

    cout<<"ingrese el valor de x: ";
    cin>>x;
    cout<<"Ingrese el valor de y: ";
    cin>>y;
    cout<<"Ingrese el valor de terminos N: ";
    cin>>N;

    for (int i=1; i<=N; i++){
        if (i%2==1){
            termino = (2*x^6*y)/(x*factorial(3*i));
        }
        else{
            termino = (x*factorial(6*i))/(5*y);
        }
        if (i%2==0){
            Sxy -= termino;
        }
        else{
            Sxy += termino;
        }
    }
}
  
```



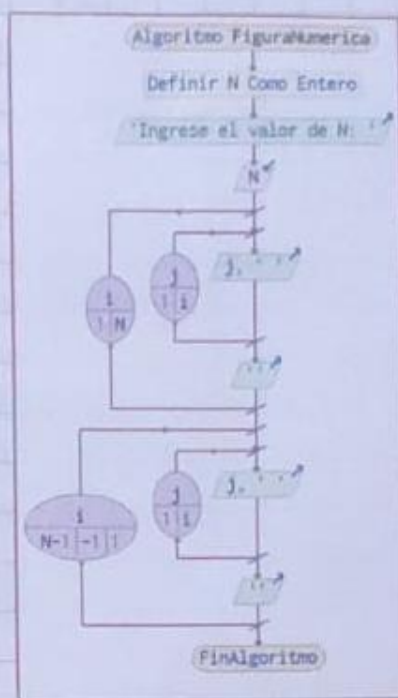
- 13) Construye un programa que, al recibir como dato un número entero positivo, escriba una figura como la que se muestra a continuación  
 $N=6$

### ANÁLISIS

**Entrada:** el usuario ingresa un número entero positivo  $NNN$

**Proceso:** Se imprimen filas con números consecutivos comenzando del 1 hasta el  $NNN$  en la primera mitad y luego descendiendo en la segunda mitad

**Salida:** se muestra la figura con las numeras en el formato especificado



### CODIGO

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

```
int main() {
    int N;
```

```
    cout << "Ingrese el valor de N: ";
    cin >> N;
```

```
    for (int i = 1; i <= N; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
```

```
    for (int i = N - 1; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
```

```
    return 0;
}
```

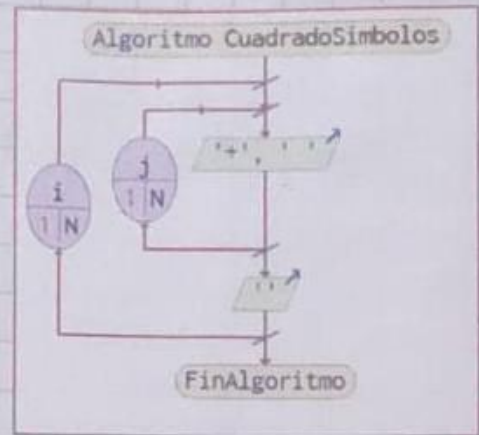
14) Realizar un programa en C++, que permita visualizar la siguiente figura (cuadrado)

### ANALISIS

**Entrada:** No se necesita entrada del usuario ya que la figura siempre es de  $3 \times 3$

**Proceso:** El programa imprime 3 filas cada una con símbolos "+"

**Salida:** Se muestra la figura de un cuadrado de  $3 \times 3$  con el símbolo "+"



### CODIGO

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

int main () {
    for (int i = 1; i <= 3; i++) {
        for (int j = 1; j <= 3; j++) {
            cout << endl;
        }
        return 0;
    }
```



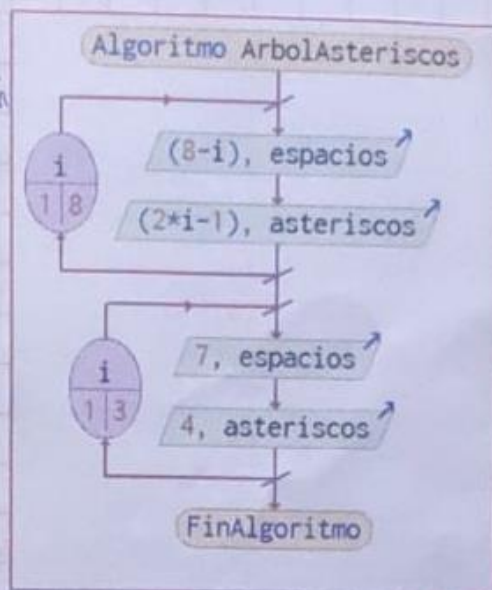
15) Construye un programa que genera la siguiente figura (árbol)

## ANÁLISIS

**Entrada:** No se requiere la configuración del usuario ya que el árbol tiene una figura fija.

**Proceso:** Se imprime una figura de árbol con varios asteriscos donde la cantidad de asteriscos aumenta en cada fila hasta obtener la copa y se imprimen dos líneas más para el tronco.

**Salida:** La figura del árbol se muestra en la figura.



## CODIGO

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

```
int main () {
    int altura, tronco;
    cout << "Ingrese la altura del árbol " << endl;
    cin >> altura;
    cout << "Ingrese la altura del árbol " << endl;
    cin >> tronco;
```

```
    for (int i=1; i<= altura; i++) {
        for (int j= altura-i; j>0; j--) {
            cout << " ";
        }
        for (int j=1; j<= 2*i-1; j++) {
            cout << " ";
        }
        cout << endl;
    }
    for (int i=0; i< tronco; i++) {
        for (int j=0; j< altura-2; j++) {
            cout << " ";
        }
        cout << " *** " << endl;
    }
    return 0;
}
```