

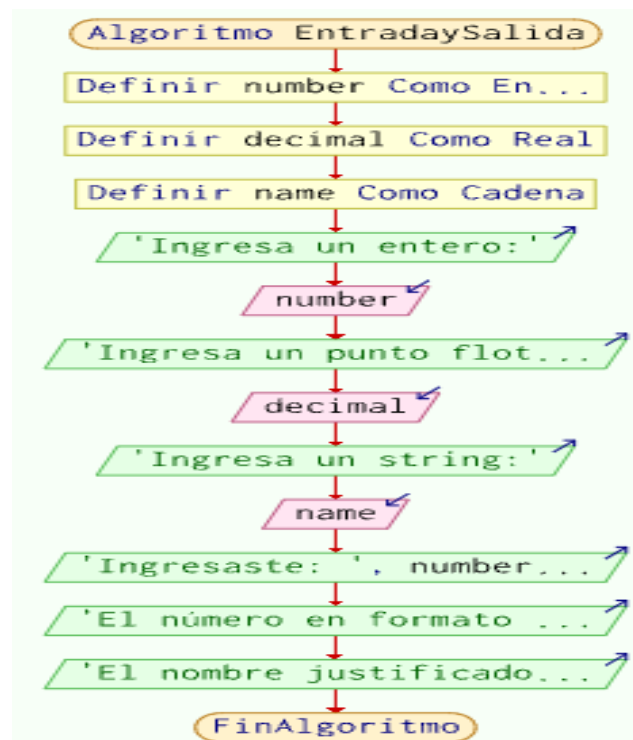
Ejercicio 1:

```
main.cpp
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int number;
7     float decimal;
8     string name;
9
10    cout << "Ingresa un entero: ";
11    cin >> number;
12
13    cout << "Ingresa un punto flotante: ";
14    cin >> decimal;
15
16    cout << "Ingresa un string: ";
17    cin >> name;
18
19    cout << "Ingresaste: " << number << ", " << decimal << ", " << name<<endl;
20
21    //Para imprimir un número de punto flotante en formato científico, se puede usar la siguiente instrucción:
22    cout << fixed << setprecision(2) << decimal << endl;
23
24    //Para imprimir una cadena en formato de título, se puede usar la siguiente instrucción:
25    //Para que sirva setw?
26    cout << setw(20) << left << name << endl;
27
28    return 0;
29 }
30
```

¿Qué Realiza el siguiente programa?

R=Este código muestra cómo solicitar entradas al usuario y cómo formatear y mostrar valores en la consola utilizando la biblioteca estándar de C++. Además, demuestra el uso de la biblioteca `*iomanip*` para controlar el formato de salida de números de punto flotante y cadenas.

Código Representado en Un Diagrama De Flujo:



Ejercicio 2:

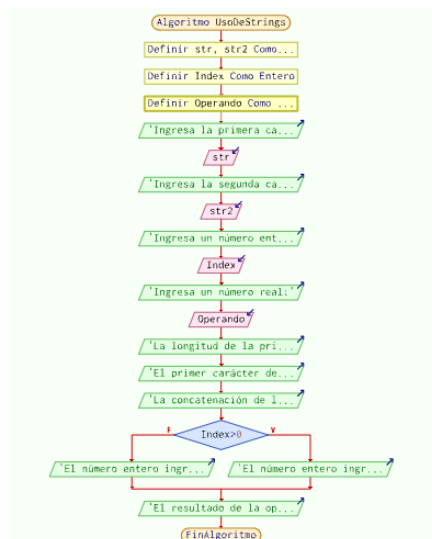
```
main.cpp
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  using namespace std;
5
6  int main() {
7      string str = "Hola mundo";
8
9      // Longitud de la cadena
10     cout << "La longitud de la cadena es: " << str.length() << endl;
11
12     // Índice de caracteres
13     cout << "El carácter en la posición 0 es: " << str[0] << endl;
14
15     // Subcadenas
16     cout << "La subcadena desde la posición 1 hasta la 3 es: " << str.substr(1, 3) << endl;
17
18     // Operaciones de comparación
19     if (str == "Hola mundo") {
20         cout << "Las cadenas son iguales" << endl;
21     } else {
22         cout << "Las cadenas son diferentes" << endl;
23     }
24
25     // Operaciones de asignación
26     string str2 = str;
27     cout << "La cadena str2 es: " << str2 << endl;
28
29     // Operaciones de transformación
30     transform(str.begin(), str.end(), str.begin(), ::toupper);
31     cout << "La cadena convertida a mayúsculas es: " << str << endl;
```

¿Para Que Sirve El Programa?

Este programa en C++ demuestra una serie de operaciones comunes que se pueden realizar con cadenas de caracteres utilizando la clase `std::string`. Primero, muestra la longitud de la cadena y el carácter en la posición 0. Luego, genera y muestra una subcadena desde la posición 1 hasta la 3.

Luego, compara la cadena con "Hola mundo" para determinar si son iguales. A continuación, realiza una asignación de cadena y muestra la cadena asignada. Posteriormente, transforma la cadena a letras mayúsculas utilizando `transform` y muestra el resultado. Luego, busca la posición de la subcadena "mundo" en la cadena y muestra la posición si se encuentra. Finalmente, itera a través de cada carácter de la cadena y los muestra uno por uno. En resumen, este programa ilustra varias operaciones de manipulación, transformación y búsqueda de cadenas de caracteres, así como la iteración a través de sus caracteres

Diagrama de Flujo



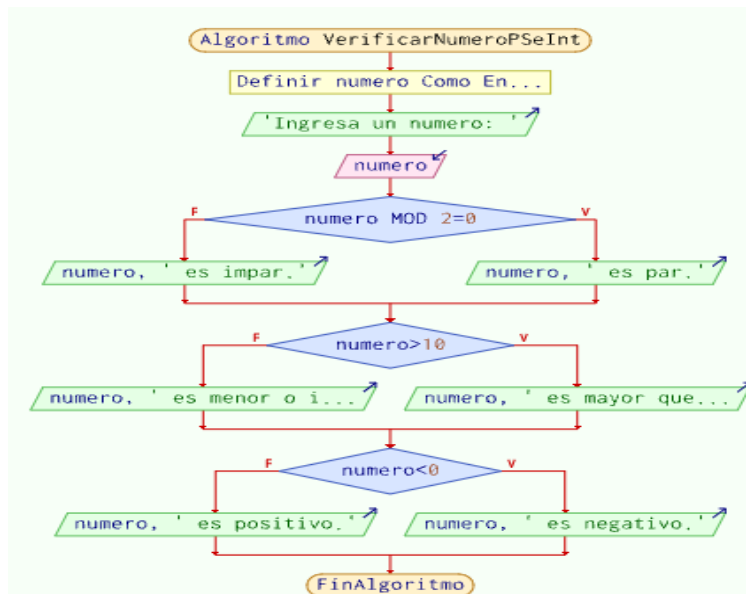
Ejercicio 3:

```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int numero;
7      cout << "Ingresa un numero: ";
8      cin >> numero;
9
10     if (numero % 2 == 0) {
11         cout << numero << " es par." << endl;
12     } else {
13         cout << numero << " es impar." << endl;
14     }
15
16     if (numero > 10) {
17         cout << numero << " es mayor que 10." << endl;
18     } else {
19         cout << numero << " es menor o igual a 10." << endl;
20     }
21
22     if (numero < 0) {
23         cout << numero << " es negativo." << endl;
24     } else {
25         cout << numero << " es positivo." << endl;
26     }
27
28     return 0;
29 }
30
```

¿Qué reliza el siguiente Programa?

R=Este código en C++ solicita al usuario que ingrese un número y luego realiza una serie de comprobaciones y declaraciones para determinar y mostrar si el número es par o impar, si es mayor que 10 o no, y si es positivo o negativo. El programa primero captura el número ingresado por el usuario utilizando la función `cin`, luego realiza las verificaciones correspondientes y muestra mensajes en función de los resultados de estas verificaciones. En resumen, el código determina y presenta información sobre la paridad, la magnitud y el signo del número ingresado por el usuario.

Diagrama De Flujo.



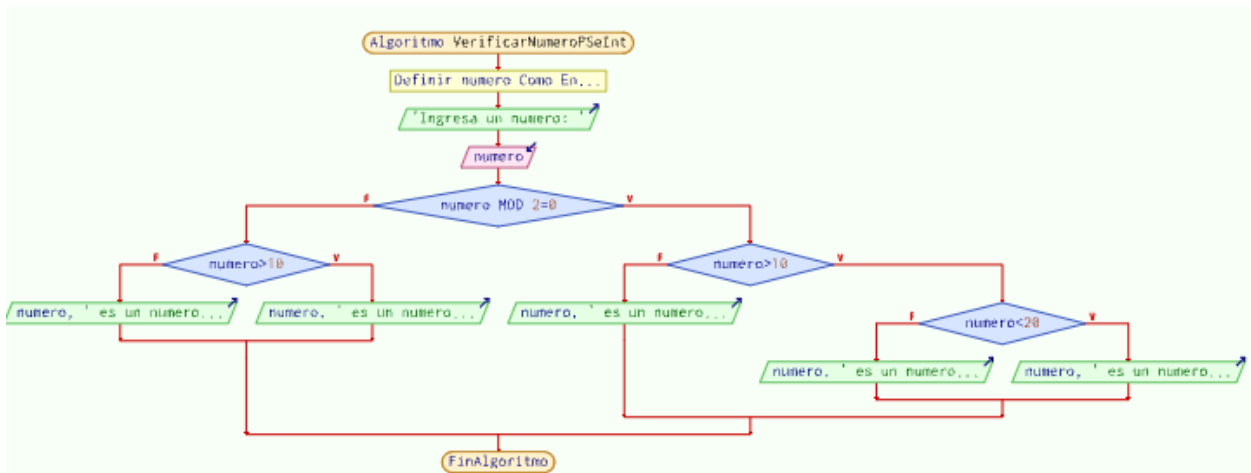
Ejercicio 4:

```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int numero;
7      cout << "Ingresa un numero: ";
8      cin >> numero;
9
10     if (numero % 2 == 0) {
11         if (numero > 10) {
12             if (numero < 20) {
13                 cout << numero << " es un numero par entre 10 y 20." << endl;
14             } else {
15                 cout << numero << " es un numero par mayor o igual que 20." << endl;
16             }
17         } else {
18             cout << numero << " es un numero par menor o igual que 10." << endl;
19         }
20     } else {
21         if (numero > 10) {
22             cout << numero << " es un numero impar mayor que 10." << endl;
23         } else {
24             cout << numero << " es un numero impar menor o igual que 10." << endl;
25         }
26     }
27
28     return 0;
29 }
30
```

¿Para que sirve el programa?

Este programa en C++ solicita al usuario que ingrese un número y luego determina si ese número es par o impar, y si es mayor, igual o menor que 10. La estructura anidada de las declaraciones condicionales permite verificar todas estas condiciones y mostrar mensajes correspondientes en función de las propiedades del número ingresado. Si el número es par, se verifica si es mayor que 10 y si es menor o igual a 20, y se muestra el mensaje adecuado. Si el número es impar, se verifica si es mayor que 10 y se muestra el mensaje correspondiente. En resumen, el programa evalúa múltiples condiciones sobre el número ingresado y proporciona información detallada sobre su paridad y su relación con el valor 10.

Diagrama de Flujo



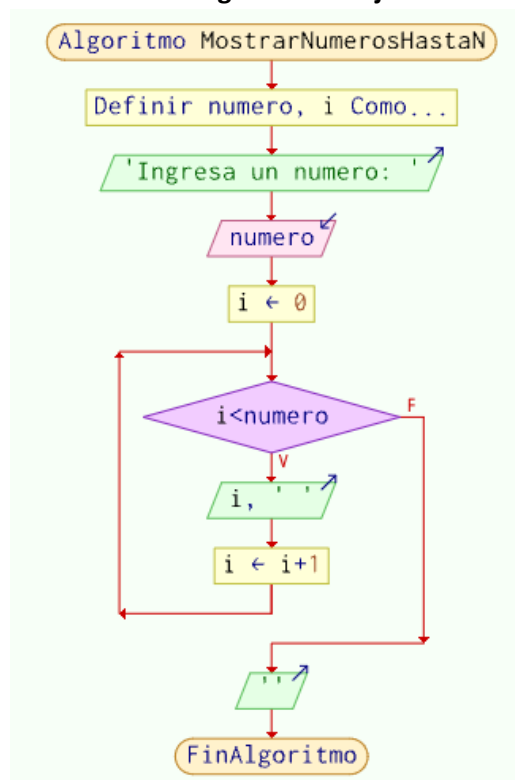
Ejercicio 5:

```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int numero;
7      cout << "Ingresa un numero: ";
8      cin >> numero;
9
10     for (int i = 0; i < numero; i++) {
11         cout << i << " ";
12     }
13
14     cout << endl;
15
16     return 0;
17 }
18
```

¿Para que sirve el siguiente programa?

Este programa en C++ tiene la finalidad de solicitar al usuario que ingrese un número y luego genera una secuencia de números desde 0 hasta el número ingresado (excluyéndolo) utilizando un bucle `for`. Cada número de la secuencia es impreso en la salida estándar, seguido por un espacio. Al final del proceso, se agrega un salto de línea para separar la salida del programa. En resumen, el programa crea y muestra una serie de números consecutivos desde 0 hasta el valor ingresado por el usuario, utilizando un bucle `for`.

Diagrama de Flujo



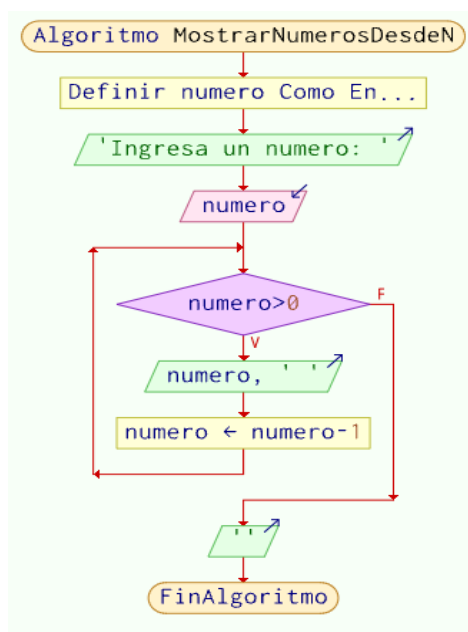
Ejercicio 6:

```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int numero;
7      cout << "Ingresa un numero: ";
8      cin >> numero;
9
10     while (numero > 0) {
11         cout << numero << " ";
12         numero--;
13     }
14
15     cout << endl;
16
17     return 0;
18 }
19
```

¿Para que sirve el siguiente programa?

Este programa en C++ tiene como objetivo solicitar al usuario que ingrese un número y luego genera una secuencia descendente de números, comenzando desde el valor ingresado y decrementando en cada iteración. Esto se logra utilizando un bucle `while` que se ejecuta mientras el valor del número ingresado sea mayor que 0. En cada iteración del bucle, se imprime el número actual y luego se reduce en 1. Una vez que el número llega a 0 o es menor, el bucle se detiene y se agrega un salto de línea para separar la salida del programa. En resumen, el programa muestra una secuencia descendente de números, comenzando desde el valor ingresado por el usuario y disminuyendo de uno en uno, utilizando un bucle `while`.

Diagrama De Flujo



Ejercicio 7

```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      char opcion;
7      cout << "Ingresa una opcion: ";
8      cin >> opcion;
9
10     switch (opcion) {
11         case '+':
12             cout << "Has ingresado la suma." << endl;
13             break;
14         case '-':
15             cout << "Has ingresado la resta." << endl;
16             break;
17         case '*':
18             cout << "Has ingresado la multiplicacion." << endl;
19             break;
20         case '/':
21             cout << "Has ingresado la division." << endl;
22             break;
23         default:
24             cout << "Opcion no valida." << endl;
25     }
26
27     return 0;
28 }
29
```

¿Para qué sirve el programa?

Este programa en C++ tiene como propósito solicitar al usuario que ingrese un carácter representando una opción matemática (+, -, *, /) y luego utiliza una estructura de control `switch` para determinar qué operación se eligió. Dependiendo del carácter ingresado, el programa muestra un mensaje correspondiente a la operación matemática seleccionada (suma, resta, multiplicación o división). Si se ingresa un carácter que no corresponde a ninguna de estas opciones, el programa muestra un mensaje indicando que la opción no es válida. En resumen, el programa identifica y muestra un mensaje basado en la opción matemática ingresada por el usuario utilizando una estructura de control `switch`.

Diagrama de Flujo

