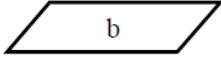
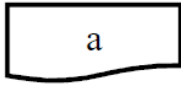
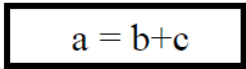
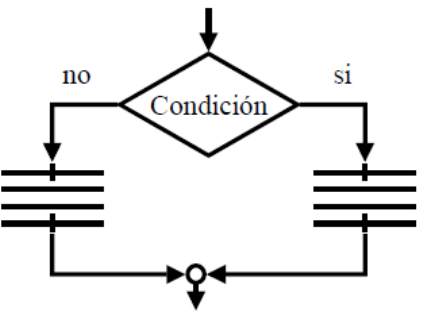
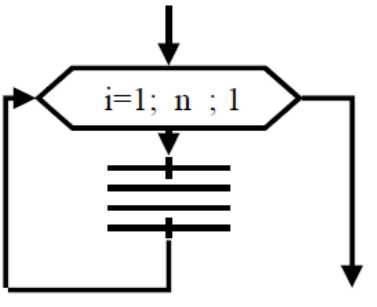
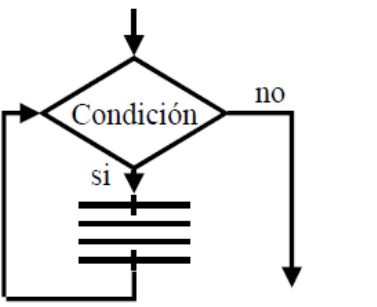
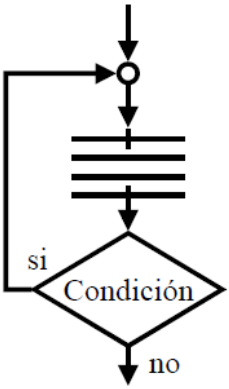


Estructuras de control			
Nombre	Diagrama de flujo	Seudocódigo	En C
Entrada o captura de datos		Leer b	scanf ( "%d", &b );
Salida o impresión de datos		Imprimir a	printf ( "cadena de formateo ", a );
Asignación		a = b + c	a = b + c ;

	<p>Si condición entonces  .....Instrucciones para  .....cuando la condición  .....sea verdadera</p> <p>Sino  .....Instrucciones para  .....cuando la condición  .....sea falsa</p> <p>Fin si</p>	<pre>if ( condición ) { .....Instrucciones en c para .....cuando la condición .....sea verdadera } else { .....Instrucciones en c para .....cuando la condición .....sea falsa };</pre>
--	--	---

<p>Estructura repetitiva Para</p>		<p>Para i=1 hasta n inc 1 haga  ..... Instrucciones a repetir  fin para</p>	<pre>for ( i=1 ; i&lt;=n ; i=i+1 ){ ..... Instrucciones a repetir };</pre>
---------------------------------------	---	---	--

<p>Estructura repetitiva Mientras que</p>		<p>MQ condición haga  ..... Instrucciones a repetir  fin MQ</p>	<pre>while ( condición ) { ..... Instrucciones a repetir };</pre>
---	---	---	---

Estructura repetitiva Hacer Mientras		Hacer ..... Instrucciones a repetir MQ condición	Do { ..... Instrucciones a repetir } while ( condición );
---	---	--	---

### Algunos operadores

---

<b>De relación:</b>	menor que:	<	<b>Aritméticos:</b>	suma:	+
	mayor que:	>		resta:	-
	menor o igual que:	<=		multiplicación:	*
	mayor o igual que:	>=		división:	/
<b>De igualdad:</b>	igual a:	==	resto división entera:	%	
	distinto de:	!=	(unario) incremento:	++	
			(unario) decremento:	--	
<b>Lógicos:</b>	(unario) negación:	!	asignación:	=	
	y (AND) lógico:	&&			
	o (OR) lógico:				

---

## FORMATOS PARA EL PRINTF

**Se escriben entre comillas.**

**Pueden contener:**

-Caracteres y secuencias de escape como "\n" (salto de línea)

**"INDICE\n"**

-Especificaciones de conversión:

Empiezan con "%" y lleva un código de conversión y un modificador opcional:

**%d %4d %7.4f**

Con el modificador se determina la presentación (longitud del campo, número de dígitos decimales, etc...)

Con el código de conversión se indica el tipo de transformación:

- d** se convierte a notación decimal
- o** se convierte a formato octal sin signo
- x** se convierte a formato hexadecimal sin signo
- f** se considera que el argumento es de tipo "float" y se convierte a notación decimal
- e** se considera que el argumento es de tipo "float" y se convierte a notación exponencial
- c** se considera que el argumento es un simple carácter
- s** se considera que el argumento es una cadena de

## FORMATOS PARA EL SCANF

### La función "scanf"

Permite leer datos con formato por el dispositivo de entrada estándar.

Lee caracteres y los convierte a representaciones internas de variables según las especificaciones de formato.

Admite parámetros de dos clases:

Especificaciones de formato  
Apuntadores a variables

```
scanf ( "%d%d", &n1, &n2 );
```

### Las especificaciones de formato:

- se escriben en tre comillas
- empiezan con el símbolo " % "
- pueden contener códigos de conversión y modificadores de conversión

Algunos códigos de conversión:

<b>d</b>	a entero decimal (int)
<b>f</b>	a coma flotante (float)
<b>c</b>	a carácter (char)