Diagramas de flujo y estructuras de control condicionales

Fundamentos de Programación

Fundamentos de Programación I

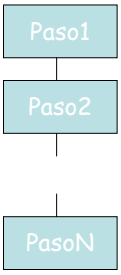
Estructuras de control y DFDs

- El orden de las operaciones dentro de un algoritmo es muy importante.

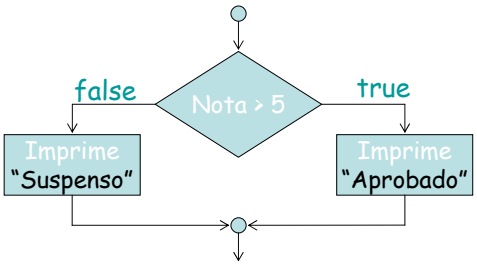
La especificación del orden = **CONTROL** del programa

Estructuras de **CONTROL** de nuestros programas

Secuenciales: Las instrucciones se ejecutan una detrás de otra

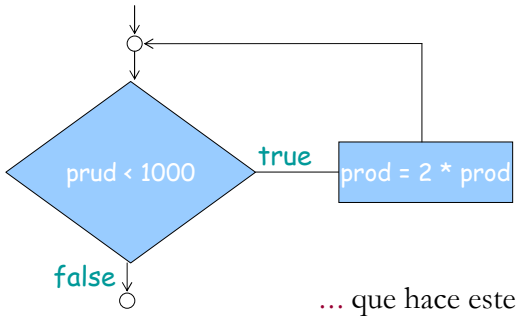


Selección (condicionales): Se toman decisiones dentro del algoritmo
(romboide = decisión)



Estructuras de **CONTROL** de nuestros programas

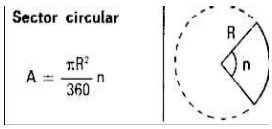
Repetición : Repetir un proceso … ¿cuantas veces?



… que hace este DFD?

Ejemplo de algoritmo secuencial

Ejercicio 1: Escribir un programa que calcule el área total o parcial (área del sector circular) de una circunferencia,



- Leer el radio del circulo (R)
- Introducir el ángulo en grados del sector (n)
- Realizar el cálculo del área según la expresión
- Mostrar el resultado del área calculada (A)

```
#include <iostream>

using namespace std;

int main()
{

//Declaracion de variables
float radio;
float angulo;
float Area;
Const float PI= 3.141516;

//lectura de datos
cout << "Introduce el radio del circulo: "<< endl;
cin >> radio;

cout << "Introduce el angulo del sector circular:" << endl;
cin >> angulo;

//Calculo del resultado

Area = PI * radio * radio *angulo / 360;

//Muestro resultados

cout << " El area del sector circular es:" << Area;


 return 0;
}
```

Condicionales en C++ (I)

```
if  ( expresió )
{
    sentencia 1
    sentencia 2
     …
}
else
{
     …
    sentencia q
     …
}
```

```
if ( expresió )
   sentencia;
else
     altra_sentencia;
```

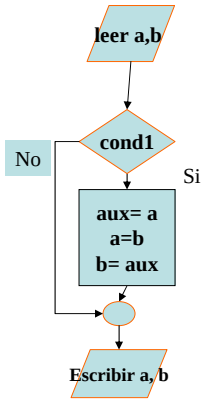
Alternativas simples

```
switch (expresión)
{
   case 1:
            sentencia1;
            sentencia2;
            break;
   case 2:
            otra_sentencia;
            break;
   default:
            otra_mas;
};
```

Alternativa multiple

- **Ejercicio 1:** Dados dos números reales intercambiarlos si no están ordenados.

  Entradas: dos números reales (a, b)
  Salidas: Dos números reales ordenados de forma ascendente
  Análisis: Se comparan y si el segundo es menor se intercambian.
  ¿Cómo se puede intercambiar el valor de dos variables?

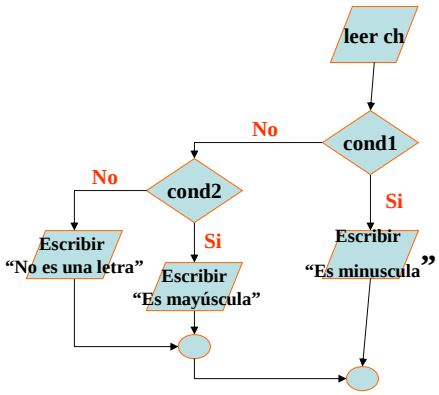


¿Están desordenados?
cond1= a > b

```
#include <iostream>

using namespace std;

int main()
{
   //Declaración de variables
   double a, b;
    double aux; //para hacer intercambio

   //Lectura de datos
   cout << "Introduce dos numeros reales: " << endl;
    cin >> a >> b;

   if ( a > b ) //Estan desordenados
   {
          aux = a;
          a = b;
          b = aux;
   }
   // else, no hay nada que hacer
    //Se muestra resultado
   cout << a << " " << b << endl;
   return 0;

}
```

- **Ejercicio 2:** Hacer un programa que indique si un cáracter es una letra mayúscula o minúscula.

  Entradas: un carácter
  Salidas: el tipo (may | min | no_es_caracter)
  Análisis: ¿Cómo se si es mayúscula o minúscula? Tabla ASCII



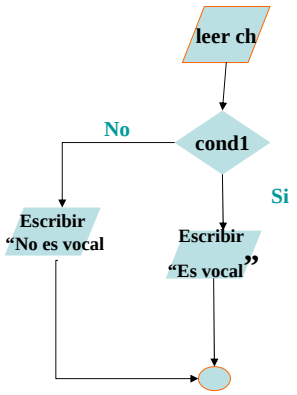
¿Es minúscula?
cond1= ch >= 'a' && ch =< 'z'

¿Es mayúscula?
cond2= ch >= 'A' && ch =< 'Z'

```
#include <iostream.h>

int main()
{
  //Declaración de variables
   char ch;

   cout << “Introduce una letra: “ << endl;
   cin >> c;

   if ( c >= ‘a’ && c =< ‘z’) //Es minuscula
         cout << “La letra : “ << c << “ es minuscula\n”;

   else if ( c >= ‘A’ && c =< ‘Z’) //Es mayuscula
          cout << “La letra : “ << c << “ es mayuscula\n”;
   else
         cout << “El carácter introducido no es una letra del
                          alfabeto.”;

  return 0;


}
```

- Ejercicio 3: Hacer un programa que indique si un carácter es vocal o consonante.

Entradas: un carácter
Salidas: el tipo (vocal| no_vocal)
Análisis: ¿Cómo se si es vocal o no? Tabla ASCII



¿Es vocal?
cond1= ch == ‘a’ || ch ==‘e’
ch==‘i’ || ch==‘o’ || ch==‘u’

```
#include <iostream>

using namespace std;

int main()

{
  //Declaración de variables
   char ch;

  //Lectura de datos
   cout << "Introduce una letra: " << endl;
   cin >> ch;

   //Toma de deciones
   if ( ch == 'a' || ch =='e' || ch='i' || ch=='o' || ch=='u')
         cout << "La letra : " << ch << " es vocal\n";

   else
         cout << "El carácter: " << ch << "  no es una vocal\n";


  return 0;
}
```

- Ejercicio 4: Realizar un programa que nos diga el porcentaje aplicado a los comerciales de una empresa para obtener las comisiones dependiendo de la categoría profesional:

Categoria	Porcentaje
A	2%
B	8%
C	12%

Entradas: Categoria profesional (cat)
Salidas: el porcentaje que se aplica a sus ventas (porc)
Análisis: ?????

```
#include <iostream>

using namespace std;

int main()
{
 char cat;
 int porc;
//Lectura de datos
 cout << "Dime la categoría profesional del comercial:"<< endl;

 cin >> cat;
//Procesamiento con condicionales
 if  (cat == 'A')
    porc = 2;
  else if (cat == 'B')
      porc = 8;
    else if (cat =='C')
          porc =12;
 //Se muestra resultado
 cout << "El porcentaje para calcular la comision de la categoria " << cat  <<
     " es :" << porc;

 return 0;
}
```

- Ejercicio 5: La empresa anterior nos propone una ampliación para que, en función de la categoría y las unidades vendidas, calcule las comisiones a sus vendedores. Estos cálculos los realiza utilizando la siguiente tabla comparativa:

Categoria	comision
Menos de 100	Ventas*100*porc_cat*2
Entre 100 y 199	Ventas*100*porc_ca*3
Mas de 199	Ventas*100*porc_cat*4

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

int main()
{
 char cat;
 int porc;

 cout << “Dime la categoría profesional del comercial:”<< endl;
 cin >> cat;

 if  (cat == ‘A’)
    porc = 2;
  else if (cat == ‘B’)
      porc = 8;
    else if (cat ==‘C’)
          porc =12;
cout << “Dime el numero de ventas realizadas:” << endl;
cin >> n_ventas;

 if (n_ventas < 100)
    comision = n_ventas * 100 *porc *2;
 else if (n_ventas < 199)
     comision = n_ventas * 100 *porc *3;
 else
     comision = n_ventas * 100 *porc *4;
 cout << “La comision para el numero de ventas es: “ << comision << endl;
 return 0;
}
```

- **Ejercicio 6:** Escribir un programa que calcule las medias de las puntuaciones (4 enteros) obtenidas por los alumnos y visualice una calificación de acuerdo a la siguiente tabla:

Media	Calificación
90-100	A
80-89	B
70-79	C
60-69	D
0-59	E

Entradas: 4 enteros (a, b, c, d)

Salidas: La calificación obtenida (cal)

Análisis: Tendremos que calcular la media de 4 valores (media) y obtener la calificación acorde a la media anterior.

```
#include <iostream>

int main()
{  //Declaración de variables
   int n1, n2, n3, n4, media;
   char cal;

   cout << “Introduce las 4 notas: “ << endl;
   cin >> n1 >> n2 >> n3 >> n4;

  //Calculo la media
   media = (n1 + n2 + n3 + n4) / 4;
   if (media >= 90) //Es A
         cal = ‘A’;
    else if ( media >= 80 ) // Es B
          cal = ‘B’;
         else if ( media > = 70 ) // Es C
                cal = ‘C’;
              else if (media >= 60) // Es D
                 cal = ‘D’;
                   else
                      cal = ‘E’;
  // Imprimo resultado
   cout << “La calificación obtenida es: “ << cal << endl;
    return 0;
}
```

- **Ejercicio 7:** El domingo de Pascua es el primer domingo después de la primera luna llena posterior al equinocio de primavera, y se determina mediante el siguiente cálculo:

A= año % 19
B= año % 4
C= año % 7
D= (19*A + 24) %30
E = (2* B + 4 * C + 6 *D +5) % 7
N = 22 + D + E

donde N es el dia del mes (si es mayor que 31 es el mes de abril)

Entradas: El año (a)
Salidas: Dia (N) y mes (mes) del domingo de pascua
Análisis: Obtener el valor de N y calcular a partir de él ,el dia y mes de acuerdo a lo indicado.

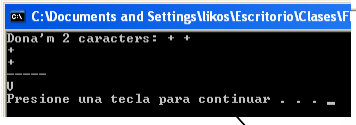
```
#include <iostream>
using namespace std;

int main()
{  int anyo, mes;//Declaración de variables
   int A, B, C, D, E, N;

   cout << “Introduce el año actual: “ << endl;
   cin >> anyo;
   //Calculos realizados
   A = anyo % 19;
   B = anyo % 4;
   C = anyo % 7;
   D= (19*A + 24) %30;
   E = ( 2* B + 4 * C + 6 *D +5) % 7;
   N = 22 + D + E;
   //Analizo N
   if ( N < = 31) // El mes es marzo
         mes = 3;
   else
   {
         N = N % 31;
         mes = 4;
   }
   cout << El domingo de pascua es: << N <<‘-’<< mes<<‘-’ << anyo;
   return 0;
}
```

- **Ejercicio 5:** En la nueva normativa de tráfico una velocidad superior al 50% a la permitida, siempre que el exceso sea mayor a 30 km /h, supone la retirada del carnet de conducir. Escribir un programa que dada la velocidad máxima permitida y la velocidad de circulación muestre si se debe retirar el carnet.

  Entradas: La velocidad máxima (vmax) y velocidad (v)
  Salidas: Indicar si se retira el carnet
  Análisis: Se comprueba si la velocidad actual es superior a la maxima, si es asi , se calcula esta diferencia y si es superior a a la mitad de la maxima y superior que 30 km/h se debe retirar el carnet. En caso contrario no.

```
#include <iostream>

int main()
{
   int v, vmax;//Declaración de variables
   int A, B, C, D, E, N;
   float tantopor;

   cout << “Introduce la velocidad de circulación y la
   maxima: “;
   cin >> v >> vmax;
   if (v > vmax)
   {
        dif = v - vmax;
        tantopor = vmax / 2;
        if (dif > tantopor && dif > 30 )
               cout << “Se debe retirar el carnet\n”;
    }
    else
        cout << “No se debe retirar el carnet\n”;
    return 0;
}
```

¿Podemos sumar dos caracteres?




```
#include <iostream>
int main(int argc, char *argv[])
{
  char a, b, res;

  cout << "Dona'm 2 caracters: " ;
  cin >> a >> b;

  res  = a + b;

  cout << a << endl;
  cout << b << endl;
  cout << "-----" << endl;
  cout << res << endl;
  system("pause");

  return 0;
}
```

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
0	0	NUL	16	10	DLE	32	20	(espacio)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	□

Tabla de códigos ASCII - Formato de caracteres estándares

Ejemplo de algoritmo secuencial

- Escribir un programa en C++ que pida dos fechas y calcule la diferencia en segundos entre ellas.

Entradas: dos fechas ¿Cómo representamos dos fechas?
Salidas: La cantidad de segundos de diferencia entre las fechas
Análisis: ¿Cómo se puede calcular la diferencia entre dos fechas?

```
#include <iostream.h>
#include <stdio.h>


int main()
{
  //Declaracion de variables
  int dd_i, mm_i, aa_i;
  int dd_f, mm_f, aa_f;
  int num_segundos;


  cout << "Introducir fecha inicial(dd mm aa): " ;
  cin >> dd_i >> mm_i >> aa_i;

  cout << endl << "Introducir fecha final (dd mm aa): " ;
  cin >> dd_f >> mm_f >> aa_f;

  //Calculo del total en segundos
  num_segundos = ((aa_f - aa_i)*365*24*60*60) + ((mm_f - mm_i)*31*24*60*60) +
         ((dd_f - dd_i)*24*60*60);
  // Salida ...


  cout << endl << "\t Transcurridos " << num_segundos << " segundos" << endl;

  return 0;
}
```

Año = 365 dias

Mes = 30 dias

Dia = 24 horas

- **Ejercicio 5:** Escribir un programa simule el comportamiento de una calculadora simple. Debe permitir sobre dos números enteros realizar las siguientes operaciones: suma +, resta -, multiplicación *, division /, y resto %.

- **Ejercicio 6:** Escribe un programa que determine el menor número de billetes y monedas de curso legal (euros) equivalente a cierta cantidad de dinero (1, 2, 5, 10, 20,50, 100, 500 euros)