SOC Servei d'Ocupació de Catalunya

nascor formación

MINISTERIO DE TRABAJO E INMIGRACIÓN

SERVICIO PÚBLICO DE EMPLEO ESTATAL

Certificado de profesionalidad IFCD0210

Generalitat de Catalunya
Departament de Treball

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Introducción

Los ordenadores requieren de programas o aplicaciones.

Es muy complicado construir programas en el lenguaje nativo de los ordenadores.

Se han creado unos lenguajes de programación que facilita esta tarea.

Lenguaje de programación es un conjunto de símbolos y reglas que combinándolos se usan para crear programas

Introducción

Los lenguajes de programación son un tipo de lenguaje:

- Léxico: símbolos permitidos y vocabulario
- Sintaxis: reglas que indican como se construye el lenguaje
- Semántica: reglas que permiten saber el significado de las expresiones que se construyen.

Introducción

Lenguaje máquina

Lenguaje ensamblador

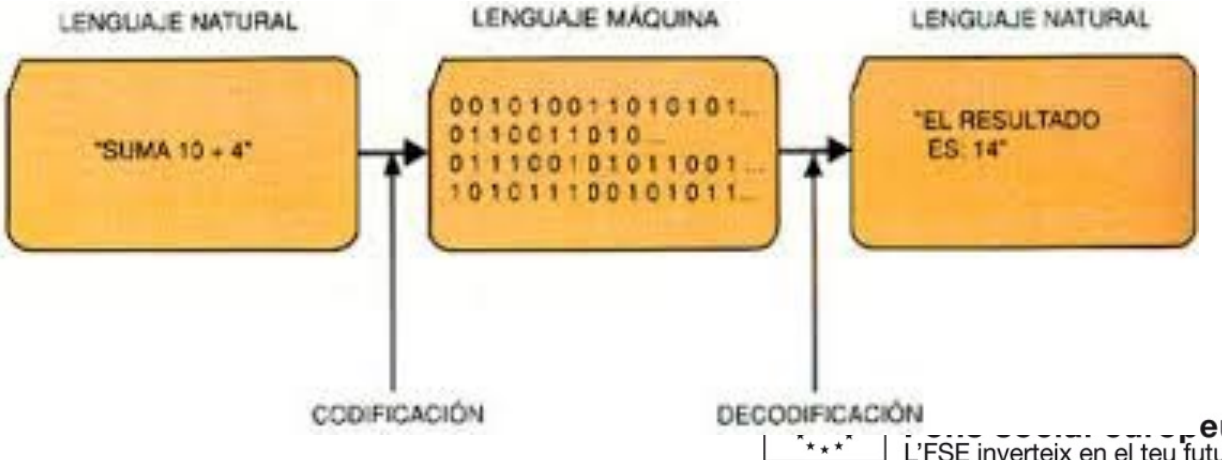
Lenguaje de alto nivel

Traductores: compiladores e interpretes

Introducción - Lenguaje máquina

•Es el lenguaje de los ordenadores.
•Estructura adaptada a los ordenadores.
•Programación difícil y requiere conocer la estructura física del ordenador.
•Se obtienen programas muy eficientes.
Características:

- •Instrucciones expresadas en binario, hexadecimal o octal.
- •Datos referenciados por las direcciones de memoria donde están.
- •Instrucciones que hacen operaciones simples
- •Instrucciones rígidas



Introducción – Lenguaje ensamblador

•Intento de hacer más humano el lenguaje de los ordenadores.
•Se traduce al lenguaje máquina con el ensamblador.
•Los programadores usan este lenguaje para realizar programas muy eficientes.
•El lenguaje continua estando ligado al procesador.

Características:

Se usa una notación simbólica para expresar los códigos de operación.

Permite un direccionamiento simbólico de variables.

Introducción - Lenguaje de alto nivel

- Se independiza el lenguaje de programación de la máquina.
- Utilizan expresiones parecidas a los humanos.
- Necesitan de un compilador o interprete.

Características:

- Independientes de la arquitectura física de la máquina.
- Requiere de un programa que traduce a LM
- Una sentencia de alto nivel da lugar a varias instrucciones de lenguaje máquina.
- Hay instrucciones potentes de uso frecuente: funciones matemáticas, E/S, tratamiento de variables, etc..
- Son menos eficientes

ejemplo C: Hola Mundo!

```
#include <stdio.h>

int main()
{
  printf("Hola Mundo!\n");
  return 0;
}
```

Introducción
Compiladores e intérpretes

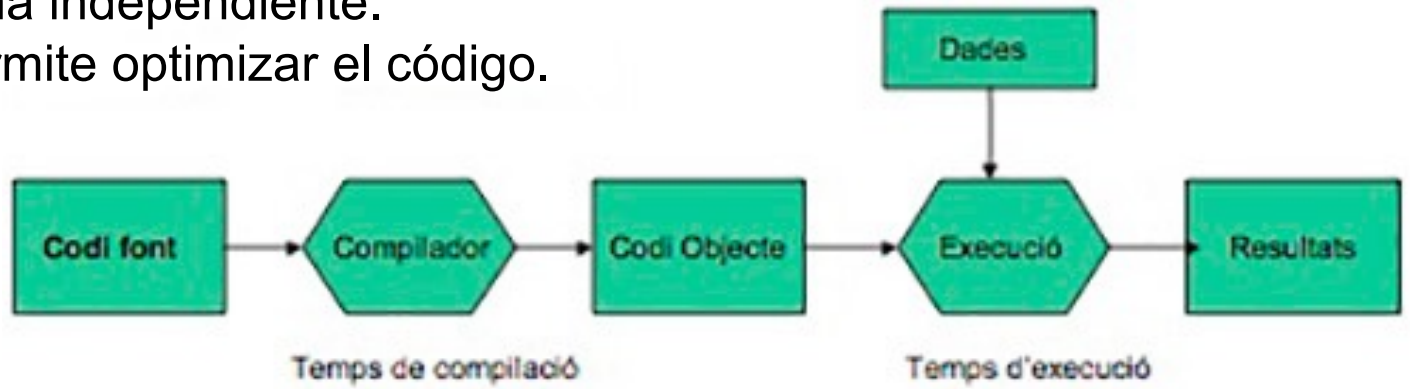
Traductor es un programa que toma como entrada un programa escrito en un lenguaje de alto nivel, llamado código fuente, y proporciona como salida otro programa semánticamente igual llamado código objeto.

Hay de dos tipos:

- Compiladores
- Intérpretes

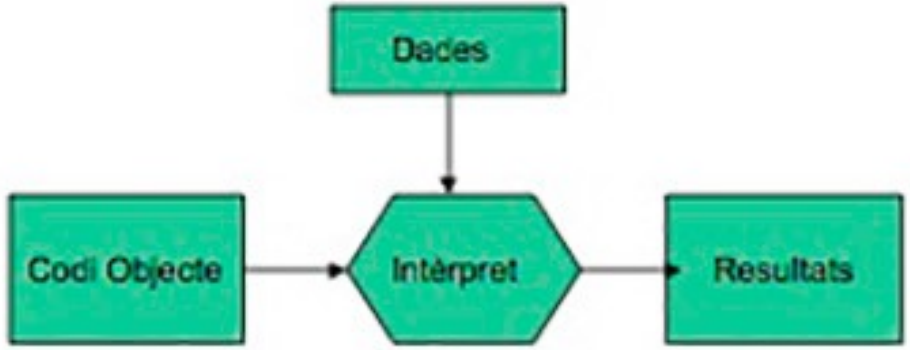
Introducción Compiladores

- Traduce el código fuente a un código objeto escrito en LM
- Durante el proceso de compilación se avisa al programador de los errores detectados y no se genera el código objeto hasta que sea un código fuente correcto.
- Los códigos fuente y objeto se guardan en dos ficheros de forma independiente.
- Permite optimizar el código.



Introducción
Intérpretes

•Permite que el código fuente se traduzca y ejecute sentencia a sentencia.
•Se usa directamente el código fuente para ejecutar programas.
•Cada vez que se ejecuta el programa se ejecuta el código.
•Es más lento y la optimización del código es la del propio código.



Introducción a los algoritmos

Fases necesarias para resolver problemas con los ordenadores:

- Planteamiento del problema
- Análisis
- Programación
- Prueba y depuración del programa
- Explotación de la aplicación

Introducción a los algoritmos

•Los programas y las sentencias se construyen de acuerdo a unas reglas y símbolos que forman parte de la gramática del lenguaje de programación.

•Los lenguajes de programación se usan para expresar los algoritmos que solucionan los problemas planteados.

•Las acciones de un algoritmo se expresan en los programas con instrucciones, sentencias o proposiciones.

•Las instrucciones básicas de todo lenguaje se pueden dividir en 4 grupos:

- Instrucciones de entrada/salida
- Instrucciones aritmético-lógicas
- Instrucciones selectivas
- Instrucciones repetitivas.

Introducción a los algoritmos

Un algoritmo se define como un conjunto ordenado de pasos a seguir para resolver un problema concreto sin ambigüedad alguna en un tiempo finito.

Mientras no suena el despertador
    Seguir durmiendo
fMientras
Levantarse
Ducharse
Desayunar
Lavarse los dientes
Vestirse
Salir de casa

Introducción a los algoritmos

```
Mientras no suena el despertador
        Seguir durmiendo
fMientras
Levantarse
SI hay agua ENTONCES
        Ducharse
fSI
SI hay desayuno ENTONCES
        Desayunar
fSI
```

```
SI hay agua ENTONCES
        Lavarse los dientes
fSI
SI ropa planchada ENTONCES
        Vestirse
SINO
        Planchar
        Vestirse
fSI
Salir de casa
```

¿Qué haceis para llegar hasta aquí?

Desde que os despertáis hasta que llegáis a la clase

¿Jugamos?

https://code.org/

https://studio.code.org/s/20-hour

Introducción a los algoritmos

0 LEER N
1 SI N=2 ENTONCES escribe “Es par”
2 SI N=1 ENTONCES escribe “Es impar”
3 SI N=2 o N=1 ENTONCES ir a 6
4 N = N – 2
5 IR a 1
6 FIN

Introducción a los algoritmos

Esquema del desarrollo de un algoritmo

- Diseño de un algoritmo
- Codificación
- Ejecución y validación

El lenguaje de los algoritmos es un conjunto pequeño y completo de sentencias mediante las cuales podemos llevar a cabo todas las acciones necesarias.

Introducción a los algoritmos

Elementos esenciales: palabras y secuencias

Restricciones y reglas:

- Constantes y programes se identifican con un nombre.
- Conjunto de instrucciones pequeño, pero completo.
- Cada sentencia se ha de escribir en una línea.
- Hay un conjunto de palabras reservadas que no se pueden usar.

Introducción a los algoritmos

Tipos de datos:

Definimos datos como cualquier objeto manipulado por el ordenador:

- Rango de datos
- Operaciones a realizar con los datos

Distinguimos dos tipos de datos:

- Constantes
- Variables

Clasificación de los tipos de datos:

- Datos simples: númericos y caracteres
- Datos compuestos: vectores y/o matrices y cadenas.

Introducción a los algoritmos

Estructuras de control

Estructura secuencial

Las instrucciones se ejecutan una detrás de la otra:

```
A = 2
A  = A -1
Escribe(A)
```

Estructura condicional

Se utilizan para tomar decisiones lógicas

```
SI A==1 ENTONCES
        Escribe('A vale 1')
SINO
        Escribe('A no vale 1')
FSI
```

Introducción a los algoritmos

Estructuras de control (cont)

Estructuras de repetición

Estructuras que se repiten varias veces

```
PARA i=2 HASTA 10
Escribe(i)
fPARA

A=1
MIENTRAS a<5 HACER
    a=a+1
    Escribe(a)
fMIENTRAS
```

Introducción a los algoritmos

```
t = 4
SI t == 5 ENTONCES
        Escribe('HOLA')
SINO
        Escribe('Adios')
fSI
```

¿Qué escribe?

Introducción a los algoritmos

```
t = 5
MIENTRAS t > = 2 HACER
        t = t – 1
        Escribe('HOLA')
FiMentre
```

¿cuántas veces se escribe HOLA?

Introducción a los algoritmos

Construir un algoritmo que pida un número y diga si es positivo o negativo

Introducción a los algoritmos

```
Escribir ("Dime un numero")
Leer x
SI x > 0 ENTONCES
        Escribir ("Es positivo")
SINO
        SI x < 0 ENTONCES
                Escribir("Es  negativo")
        SINO
                 Escribir("Es cero");
        fSI
fSI
```

Introducción a los algoritmos

Realizar un algoritmo que permita calcular el área de un rectángulo. Leer la Base y la Altura.

AREA = BASE * ALTURA

Introducción a los algoritmos

Realizar un algoritmo que permita introducir 2 números y calcular la suma y el producto de ambos.

Introducción a los algoritmos

A. Calcular y mostrar los números entre 1 y 100 que son divisibles por uno dado.

B. Calcular la suma de todos ellos.

C. Calcular la suma de los 5 primeros números.

Expresiones boolenas

Las expresiones booleanas se usan para determinar si un conjunto de una o más condiciones es verdadero o falso, y el resultado de su evaluación es un valor de verdad.

Expresión1 OPERADOR BOOLEANO Expresión2

2 < 3
A < 3
A < B

Expresiones boolenas - Operadores

= = Igual
<> O != No igual
< Menor que
<= Menor o igual que
> Mayor que
>= Mayor o igual que

Expresiones boolenas - Operadores

A == B : A igual que B
A <> B : A diferente que B
A != B : A diferente que B
A < B : A menor que B
A <= B : A menor o igual que B
A > B : A mayor que B
A >= B : A mayor o igual que B

Expresiones boolenas - Operadores

NOT (NO) Este operador produce el valor Verdadero, si su operando es Falso; y el valor Falso, si su operando es Verdadero.

AND (Y) Este operador produce el valor Verdadero si ambos operandos son Verdadero. Si cualquiera de los dos operandos es Falso, entonces el resultado será Falso.

OR (O) Este operador realiza una operación O-inclusivo. El resultado es Verdadero si cualquiera de los dos operandos, o ambos son Verdadero. En caso contrario, es Falso.

Expresiones boolenas - Operadores

VERDADERO OR FALSO = VERDADERO 1 + 0 = 1
FALSO OR VERDADERO = VERDADERO 0 + 1 = 1
VERDADERO OR VERDADERO = VERDADERO 1 + 1 = 2
FALSO OR FALSO = FALSO 0 + 0 = 0

OR = +
AND = *
V = 1 F=0

VERDADERO AND FALSO = FALSO
1 * 0 = 0
FALSO AND VERDADERO = FALSO
0 * 1 = 0
VERDADERO AND VERDADERO = VERDADERO
1 * 1 = 1
FALSO AND FALSO = FALSO
0 * 0 = 0

Expresiones boolenas - Operadores

El coche es rojo, pequeño y veloz

SI coche es rojo AND coche es grande
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

El coche es rojo, pequeño y veloz

SI coche es rojo OR coche es grande
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

Expresiones boolenas - Operadores

El coche es rojo, pequeño y veloz

SI coche es rojo AND coche es pequeño AND coche es lento
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

Expresiones boolenas - Operadores

El coche es rojo, pequeño y veloz

SI (coche es rojo AND coche es pequeño) OR coche es lento
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

Expresiones boolenas - Operadores

El coche es rojo, pequeño y veloz

SI coche es rojo AND (coche es pequeño OR coche es lento)
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

Expresiones boolenas - Operadores

El coche es rojo, pequeño y veloz

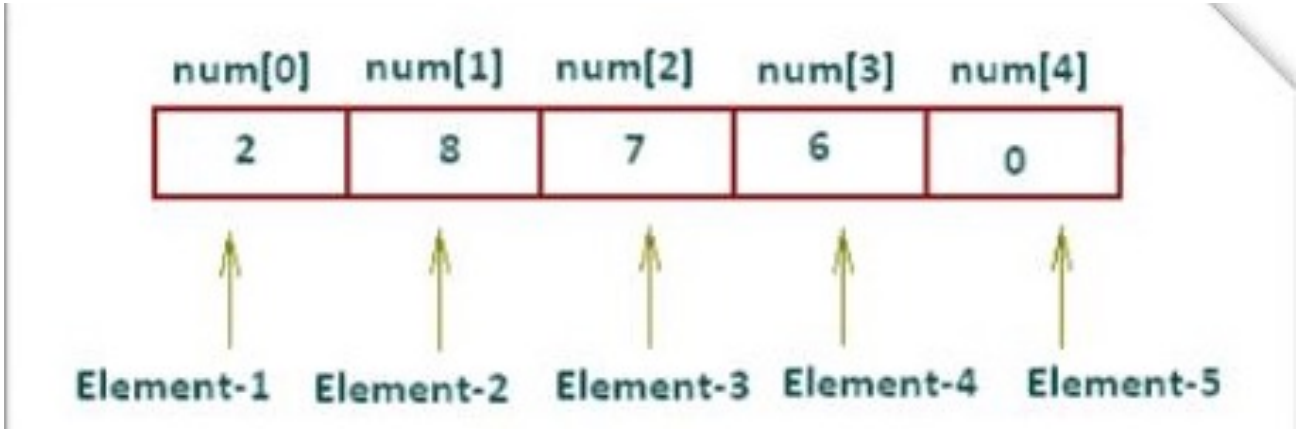
SI coche es rojo AND NOT coche es lento
ENTONCES
    Escribe('1')
SINO
    Escribe('2')
FSI

Expresiones boolenas - Operadores

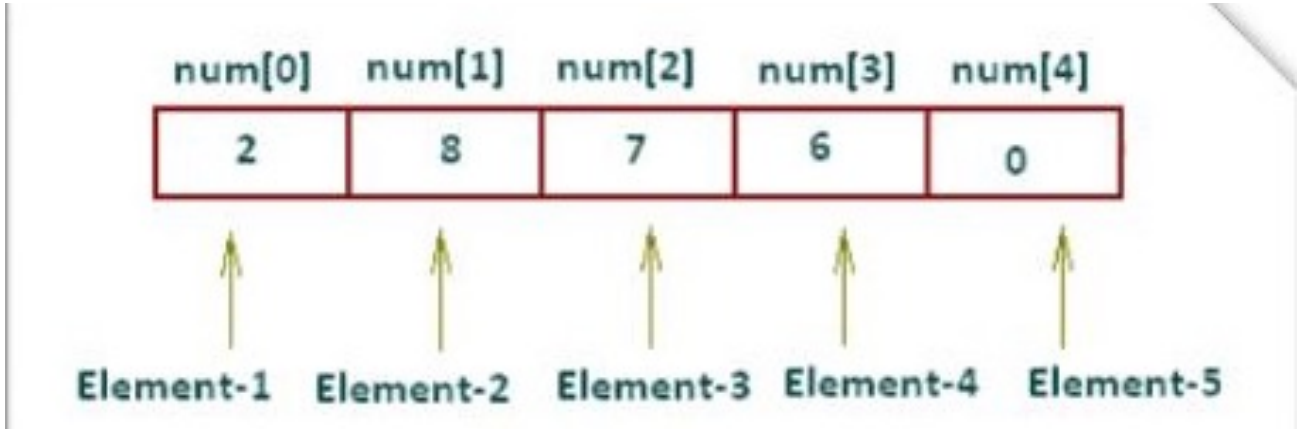
Expresión lógica	Resultado	Observaciones
(1>0) **y** (3=3)	Verdadero	La primera condición es verdadera **y** la segunda tambien es verdadera.
(0<5) **o** (0>5)	Verdadero	Con cualquiera de las dos condiciones que se cumpla el resultado será verdadero porque se usa el operador **"o"**.
(5<=7) **y** (2>4)	Falso	La primera condición es verdadera, pero la segunda es falsa, por lo tanto el resultado es falso debido a que se usa el operador **"y"**.
no (5>5)	Verdadero	La condición es falsa, pero la negación de falso es verdadero.
(número = 1) **o** (7>=4)	Verdadero	número vale 5, por lo tanto la primera condición es falsa pero la segunda es verdadera y con el operador **"o"** con una que se cumpla toda la expresión es verdadera.

Vectores / Array

Conjunto ordenado que contiene un nº fijo de elementos (su dimensión) de cualquier tipo válido definido con la condición de que todos deben ser del mismo tipo



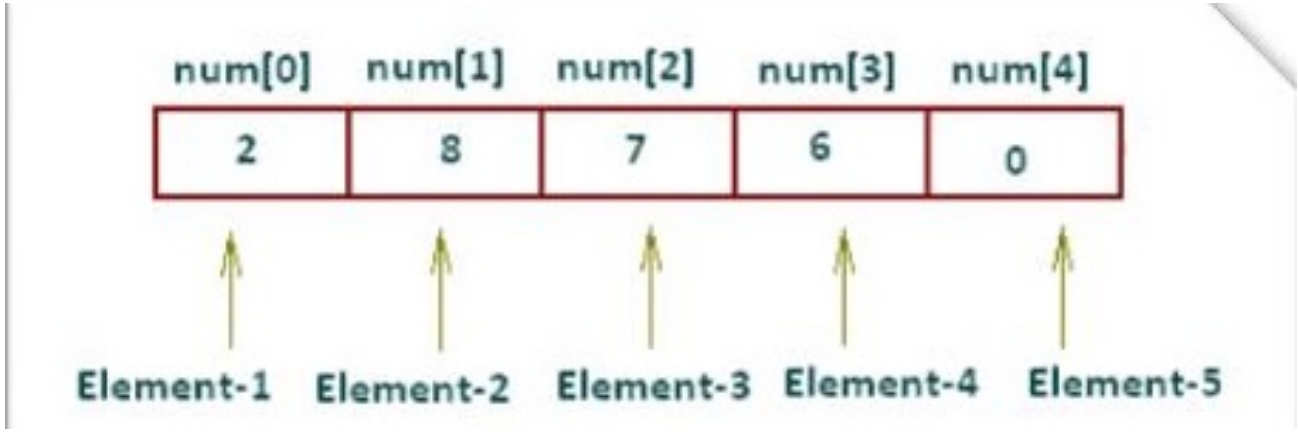
Vectores / Array



num[i] : num [0] = 2, num[1]=8, num[2]=7,

num[3] = ??? Y num[4] = ???

Vectores / Array




```
Para i=0 hasta 4 hacer
        escribir(num[i])
fPara
```

Vectores / Array

```
Para i=0 hasta 4 hacer
        num[i]=Leer(X)
fPara
```

0	1	2	3	4

Vectores / Array

Haz un algoritmo que nos pida 6 datos y luego los muestre en orden contrario al que se han introducido

SI MULTIPLE: Según

```
SEGÚN idioma HACER
        CASO ‘Español’:
                Escribir(“Hola”)
                Salir
        CASO ‘Inglés’:
                Escribir(“Hi”)
                Salir
        CASO ‘Catalán’:
                Escribir(“Hola”)
                Salir
        SINO:
                Escribir(“Hola”)
fSEGUN;
```

SI MULTIPLE: Según

Haz un algoritmo que en función del mes entrado como número devuelva el literal del mes.

CADENAS

De la misma forma que se asignan números:

A=10

También se asignan cadenas de caracteres

A=“Borja”

Escribir Nombre
Nombre=“Borja”
Escribir Nombre

EJERCICIO A1

```
<INPUT
        TYPE="TEXT"
        NAME="NOMBRE"
>

NOMBRE="Valor que pongamos en texto"

<SELECT NAME="CURSO">
        <OPTION VALUE="CURSO1">
        <OPTION VALUE="CURSO2">
        <OPTION VALUE="CURSO3">
</SELECT>

CURSO="CURSO1"
```

EJERCICIO A2

Suponemos un formulario de una academia de formación de HTML con los siguientes campos:

NAME=Curso: Curso1, Curso2, Curso3
NAME=Nombre
NAME=Teléfono
NAME=Email

Queremos que se envíe un email con esta información en función del curso a:

Curso1: curso1@alu.com
Curso2: curso2@alu.com
Curso3: curso3@alu.com

Existen las sentencias:

EnviarEmail(TO,MENSAJE)
RecibirParametroWEB(PARAMETRO)

EJERCICIO A3

Desarrollar un algoritmo que permita leer dos valores distintos y determinar cual de los dos valores es el mayor. Escribir el resultado.

Realizar un algoritmo, además, que sume los dos números.

EJERCICIO A4

Desarrolle un algoritmo que lea cuatro números distintos y determine cual de los cuatro es mayor.

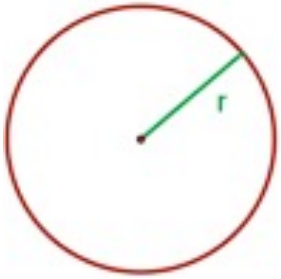
Pista: Utilizar una tabla

EJERCICIO A5

Desarrollar un algoritmo que sume los números enteros entre 1 y 100 de 2 en 2.

EJERCICIO A6

Determinar la longitud de una circunferencia y el área del circulo



$$L = 2 \cdot \pi \cdot r$$

$$A = \pi \cdot r^2$$

EJERCICIO A7

Desarrolle un algoritmo que permita convertir calificaciones numéricas a letras según la siguiente tabla de conversión:

NÚMERO	LETRA
19-20	A
16-18	B
12-15	C
9-11	D
0-8	E

EJERCICIO A8

Desarrolle un algoritmo para determinar el pago de entradas de espectáculo en función del número que compren.

NÚMERO	DESCUENTO
1	-
2	10%
3	15%
4	20%
5 o más	25%

EJERCICIO A9

Desarrolle un algoritmo que calcule el promedio de 10 notas.

EJERCICIO A10

Desarrolle un algoritmo que ordene 3 números entrados aleatoriamente.

Desarrolle un algoritmo que ordene 10 número entrados aleatoriamente. Usar el método de la burbuja.

SOC Servei d'Ocupació de Catalunya

nascor formación

MINISTERIO DE TRABAJO E INMIGRACIÓN

SERVICIO PÚBLICO DE EMPLEO ESTATAL

Certificado de profesionalidad IFCD0210

Generalitat de Catalunya
Departament de Treball

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur