Alfredo Vellido : www.lsi.upc.edu/~avellido

Fonaments d'Informàtica

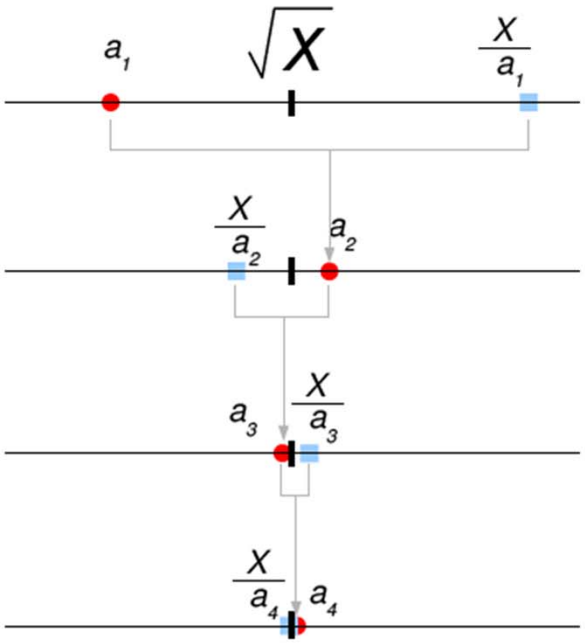
Semana 1-2: Introducción a la programación

¿Qué es un algoritmo?

- **Wikipedia dixit:** En matemáticas, ciencias de la computación, y disciplinas relacionadas, un algoritmo (del latín, *algorithmus* y éste a su vez del matemático persa *al-Jwarizmi*) **es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema**. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. Los algoritmos son objeto de estudio de la ***algoritmia***.

Ejemplo: El famoso algoritmo de Newton

Mètode de Newton



Algorisme:

1. Comencem amb una aproximació a=1.
2. Calculo l'invers de l'aproximació (*X/a*).
3. Si la distància entre els dos és menor que *E*, anar al pas 5.
4. Calculo la nova aproximació com la mitjana entre *a* i *X/a*. Anar al pas 2.
5. Final.

Características de un algoritmo

- **Carácter finito**. "Un algoritmo siempre debe terminar después de un número finito de pasos".
- **Precisión**. "Cada paso de un algoritmo debe estar precisamente definido; las operaciones a llevar a cabo deben ser especificadas de manera rigurosa y no ambigua para cada caso".

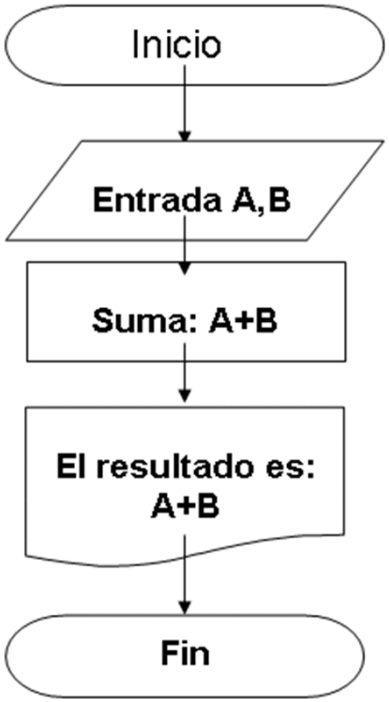
Características de un algoritmo

- **Entrada**. "Un algoritmo tiene cero o más entradas: cantidades que le son dadas antes de que el algoritmo comience, o dinámicamente mientras el algoritmo corre. Estas entradas son tomadas de conjuntos específicos de objetos".
- **Salida**. "Un algoritmo tiene una o más salidas: cantidades que tienen una relación específica con las entradas".
- **Eficacia**. "También se espera que un algoritmo sea eficaz, en el sentido de que todas las operaciones a realizar en un algoritmo deben ser suficientemente básicas como para que en principio puedan ser hechas de manera exacta y en un tiempo finito por un hombre usando lápiz y papel".

Descripción de un algoritmo

La descripción de un algoritmo se suele hacer a tres niveles:

- **Descripción de alto nivel**. Se establece el problema, en su caso un modelo matemático, y se explica el algoritmo de manera verbal, posiblemente con ilustraciones y omitiendo detalles.
- **Descripción formal**. Se usa pseudocódigo para describir la secuencia de pasos que encuentran la solución.
- **Implementación**. Se muestra el algoritmo expresado en un lenguaje de programación específico o algún objeto capaz de llevar a cabo instrucciones.



¿Qué es un lenguaje de programación?

- Un **lenguaje de programación** es un conjunto de símbolos y de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.
- Aunque muchas veces se usan los términos '**lenguaje de programación**' y '**lenguaje informático**' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el **HTML**.

Del algoritmo a la programación

¿Qué es un lenguaje de programación?

- Un lenguaje de programación permite a un programador **especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias**. **Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural.**
- Las **gramáticas formales** son objeto de estudio de la lingüística computacional. **La sintaxis de cada lenguaje de programación se define de hecho por una gramática formal.** En teoría de la **informática** y en matemática, la gramática formal define lenguajes formales.

Lenguajes de programación

Categorías de lenguaje de programación (según nivel de abstracción)

- **Lenguaje de máquina**: Están "escritos" de manera sólo directamente legible por la máquina (computadora), ya que sus instrucciones son cadenas binarias (0 y 1).
- **Lenguaje de bajo nivel**: Los lenguajes de bajo nivel son lenguajes de programación que se acercan al funcionamiento de una computadora. El lenguaje de más bajo nivel por excelencia es el **código máquina**. A éste le sigue el **lenguaje ensamblador**, ya que al programar en ensamblador se trabajan con los registros de memoria de la computadora de forma directa.

Lenguajes de programación

Categorías de lenguaje de programación (según nivel de abstracción)

- **Lenguaje de medio nivel**: Hay lenguajes de programación que son considerados por algunos expertos como lenguajes de medio nivel (como es el caso del **lenguaje C**) al tener ciertas características que los acercan a los lenguajes de bajo nivel pero teniendo, al mismo tiempo, ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.

Lenguajes de programación

Categorías de lenguaje de programación (según nivel de abstracción)

- **Lenguaje de alto nivel**: Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de "lenguajes naturales", como el inglés. En BASIC, uno de los lenguajes de alto nivel más conocidos, un comando cuasi-verbal tal como "**IF CONTADOR = 10 THEN STOP**" es utilizado para pedir a la computadora que pare si el CONTADOR toma el valor 10. **Esta forma de trabajar podría inducir a pensar que las computadoras parecen comprender un lenguaje natural; en realidad lo hacen de una forma rígida**.

Lenguajes de programación

El programa como objeto de la programación

Anomenem **programa**:

- Al codi font d'un programa
- Al fitxer executable

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

int main()
{
   int x, y;
   ...

   return 0;
}
```

Compilar

```
01000100010110100100
01010011010101011010
01010111010101101010
00101011101010101010
00101011010111101000
10101000101010001010
01010100010111101010
11001011011011011010
01011111101110101010
10101101101010100101
```

Compilar un programa significa traduir el *codi font* a *codi màquina*. Aquesta és la tasca del **compilador**.

¿Qué es un programa?

- **Algoritmo para máquinas computadoras implementado en un lenguaje de programación específico**. Fases de implementación:
  - **Construcción**: Editor de texto
  - **Compilación/Linkado**: Compilador
  - **Ejecución**: Sistema Operativo/Interprete
- Los ordenadores son tontos así que los lenguajes de programación han de ser muy rígidos:
  - **Desventajas**: Hay que conocer las normas al detalle …
  - **Ventajas**: Se pueden compilar y traducir de manera automática
- Etapas en la construcción de un programa (sencillo):

  Especificación, **diseño** e **implementación**.

Lenguajes de programación

Errores de programación

Errores de compilación

- **Hemos escrito el programa mal** y el compilador no ha sabido traducir el código fuente a un fichero ejecutable. El compilador nos muestra mensajes que se refieren a las partes del código fuente que no entiende.

Errores de ejecución

- **El programa se ha podido compilar, pero** al ejecutarlo no hace exactamente lo que se pretendía. También puede pasar que el programa haga una “operación no válida” y acabe abruptamente (el sistema operativo lo cierra a la fuerza) en ciertas condiciones.

Errores de programación: ejemplo

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

int main()
{
  int a,b,tmp
  cin >> a >> b;

  while ( b != 0 ) {
    tmp = b;
    b = a % b;
    a = tmp;
  }
  cout << a;
}
```

Ens hem oblidat **un** punt i coma aquí

```
euclides.cpp: In function 'int main()':
euclides.cpp:8: error: expected initializer before 'cin'
euclides.cpp:11: error: 'tmp' was not declared in this scope
```

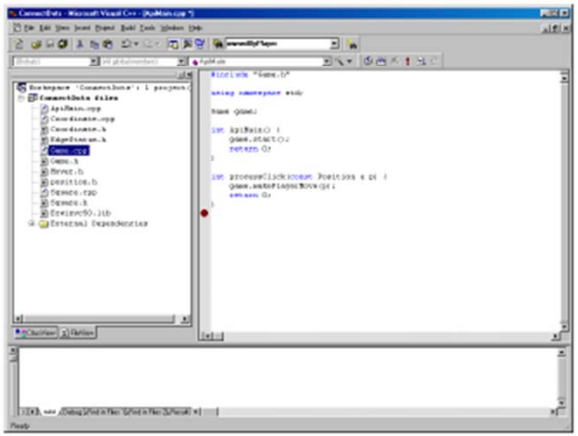
Sistema operativo (SO)

- Un **SO** es un programa o conjunto de programas que en un sistema informático (no sólo un ordenador) gestiona los recursos de hardware y provee servicios a los programas de aplicación
- Tiene prioridad respecto al resto de programas.
- El **kernel** es un software que constituye la parte más importante del SO. Tiene a cargo facilitar a los distintos programas acceso seguro al hardware de la computadora. También se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como **multiplexado**. Acceder al hardware directamente puede ser complicado, por lo que los kernel suelen usar *abstracciones* del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador**.**

Lenguajes de programación

IDE's

- Integrated Development Environments or IDEs
  - Sistemas que integran todas las componentes del ciclo de desarrollo de software
    - E.g., MS Visual C++, Borland, Code Warrior, Dev-C++
- Provee de todas las capacidades para desarrollar software
  - Editor
  - Compilador
  - Linker
  - Loader
  - Debugger
  - Visualizador



Novedad 12/13: Codeblocks!

Lenguajes de programación

La asignatura: Dev-C++ || Codeblocks

1. Etapes en el desenvolupament d'un programa

Etapa	Descripció	Resultat	Farem servir...	Llenguatge del resultat
Formalització de l'enunciat	Especificar el problema que volem resoldre.	Especificació	Llapis i paper	Llenguatge natural ó llenguatge formal.
Disseny i Implementació	Identificar l'esquema necessari i trobar un programa que resolgui el problema	Programa en C++	Llapis i paper	Llenguatge C++
Edició	Obtenir un fitxer font	Programa font en C++	Editor Dev-C++	Llenguatge C++
Compilació	Obtenir un executable	Programa executable	Compilador Dev-C++	Llenguatge màquina
Prova	Provar l'executable	Procés en execució	Depurador Dev-C++	N/A

La asignatura: Dev-C++

C++: a considerar:

`# include < …>` → librerías … los comandos no son infinitos

`endl;` → el formato importa

`system("pause")` → lo que el ojo no ve

`cin, cout` → lo que entra y lo que sale

`//  /* */` → comentando programas

`" … "` → el texto

… ejecución desde terminal o desde IDE