Universidad Carlos III de Madrid
www.uc3m.es

Lesson 1

Introduction to Computer Science and Programming

Programming

Grade in Industrial Technology Engineering

Outline

1. Computers and Computer Science
   - Computers and Computer Science
   - Brief history of computers
2. Information representation with computers
3. Algorithms and programs
   - Notion of algorithm and program
   - Tools for algorithm design
4. Computer Science in the context of Industrial Engineering

jgromero@inf.uc3m.es

Outline

1. **Computers and Computer Science**
   - **Computers and Computer Science**
   - Brief history of computers
2. Information representation with Computers
3. Algorithms and programs
   - Notion of algorithm and program
   - Tools for algorithm design
4. Computer Science in the context of the Industrial Engineer

jgromero@inf.uc3m.es

Computer Engineering (CE)

Develop computers

"…software and firmware for embedded microcontrollers, designing VLSI chips, designing analog sensors, designing mixed signal circuit boards, and designing operating systems. Computer engineers are also suited for robotics research, which relies heavily on using digital systems to control and monitor electrical systems like motors, communications, and sensors." (Wikipedia)

Computer Science (CS)

Develop **processes to create, describe, and transform information** from a *mathematical* point of view

"…is more on understanding the properties of the programs used to implement software such as games and web-browsers, and using that understanding to create new programs or improve existing ones." (Wikipedia)

Information Technology (IT)

Build and maintain **computer systems**

"…the study, design, development, implementation, support or management of computer-based information systems, particularly software applications and computer hardware." (Information Technology Association of America - ITAA)

"IT deals with the use of electronic computers and computer software to securely convert, store, protect, process, transmit, input, output, and retrieve information.

[…] IT professionals perform a variety of duties that range from installing applications to designing complex computer networks and information databases. A few of the duties that IT professionals perform may include data management, networking, engineering computer hardware, database and software design, as well as the management and administration of entire systems." (Wikipedia)

Informática (Informatics)

General denomination in Spanish

"Ciencia aplicada que abarca el estudio y aplicación del tratamiento automático de la información, utilizando sistemas computacionales, generalmente implementados como dispositivos electrónicos.

También está definida como el procesamiento automático de la información." (Wikipedia)

"Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores." (RAE)

"The science of information, the practice of information processing, and the engineering of information systems." (Wikipedia)

Computer

Electronic machine able to accept input data, to perform different kind of operations with data, and to provide the resulting information as the output.

This transformation process is controlled by a set of instructions (*the program*), which has been previously stored in the computer.

Programming

"5. tr. Inform. To develop programs to solve problems with computers." (RAE)

Provide a computer with data and instructions on what should be done with the data for the resolution of a given problem

Automatic information processing

- Information acquisition from input devices
- Information representation to be used by the computer
- Information transformation by application of logic and arithmetic operations
- Information storage in the memory of the computer
- Information transmission through output devices

Features of computers

- High speed to perform automatic information processing
- Large computation power
- Huge capacity for storing data and programs
- Able to communicate with other computers (networking) and people (human-machine interfacing)

Computer elements

- Physical support: **Hardware**

  Mechanical and electronic elements that *physically* compose the computer

  memory modules, screen, keyboard, motherboard, etc.

- Logical support: **Software**

  Logic part that directs the actions of the physical support

  Set of programs that make the computer perform useful tasks

  text processor, image editor, video game, etc.

Classification of computers:

- **Supercomputers**: Used for very intensive scientific and technical computation
- **High-performance servers**: Provide access to large datasets from distributed terminals (hundreds of thousands of concurrent users)
- **Basic servers**: Multi-user and networked applications in enterprises (dozens of concurrent users)
- **Personal computers**: Single user applications
- **Mobile computers**: Performance is reduced to increase mobility (smartphones, tablets, etc.)

Universidad Carlos III de Madrid
www.uc3m.es

Outline

1. **Computers and Computer Science**
   - Computers and Computer Science
   - **Brief history of computers**
2. Information representation with Computers
3. Algorithms and programs
   - Notion of algorithm and program
   - Tools for algorithm design
4. Computer Science in the context of the Industrial Engineer

- Multimedia resources
  - Computer History Museum
    http://www.computerhistory.org/revolution/topics
  - G. O'Reagan. *A Brief History of Computing. Springer*, 2012.

Motivation

Avoid repetitive tasks and automatic management of information

Calculations

Abacus: Chinese or Babylonian origins (3.500 B.C.)

Pascal arithmetical machine (1642)

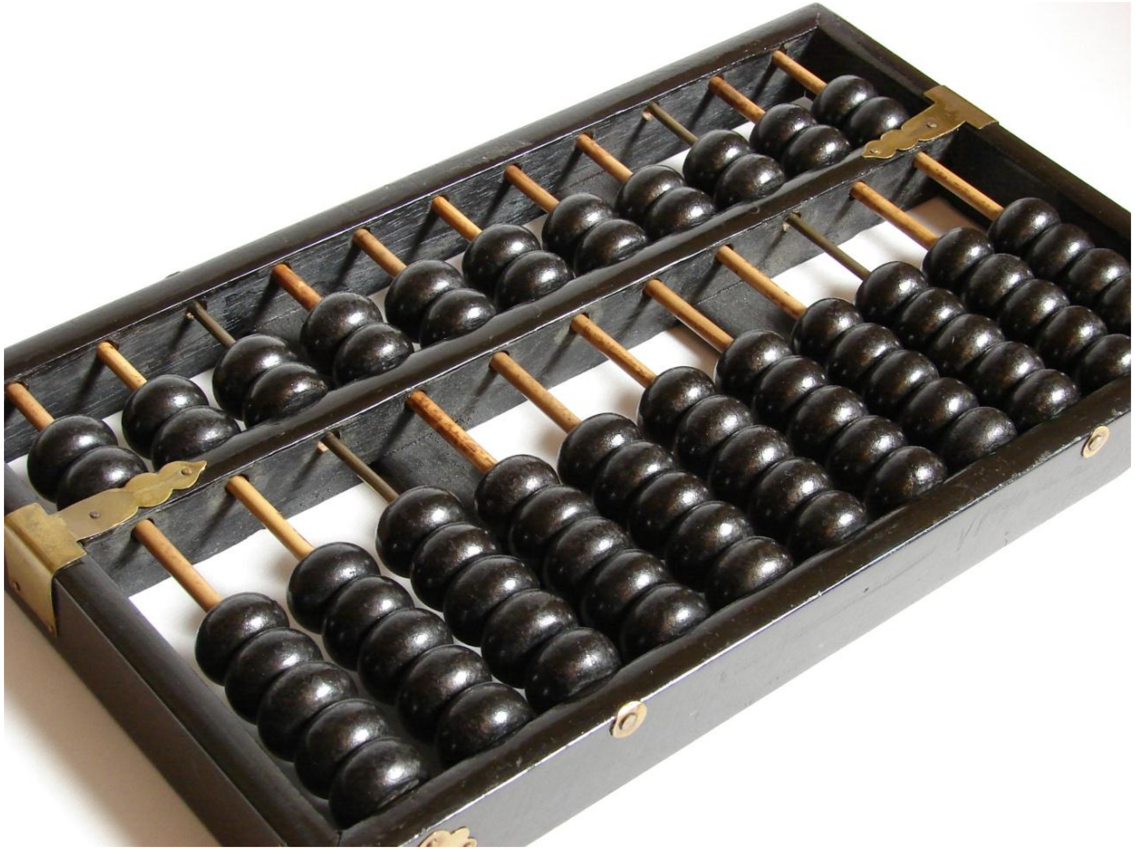
Leibniz's universal calculator (+, -, *, /, sqrt) (1671)

Task programming

Jacquard loom (1801)

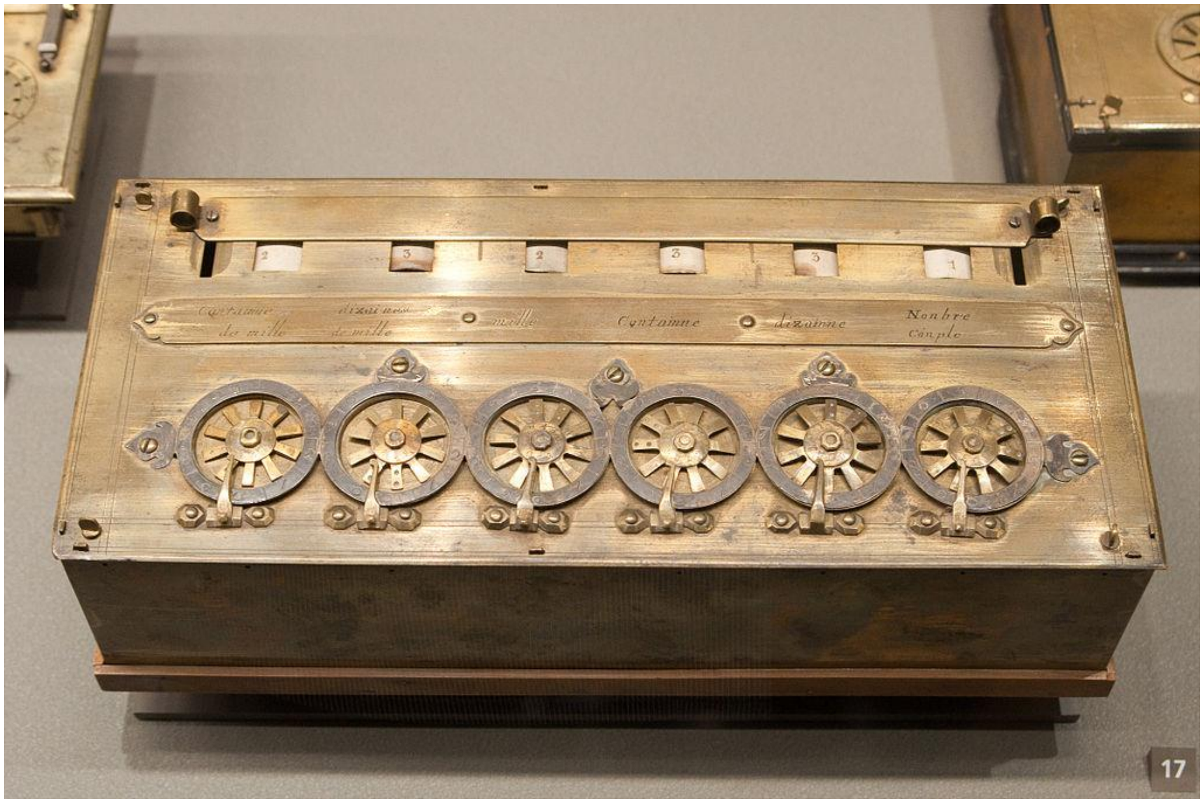
Info

1.2. Brief history of computers



Chinese abacus

Source: Wikimedia Commons [link]



Pascal arithmetical machine

Source: Wikimedia Commons [link]

Info

1.2. Brief history of computers

Jacquard loom

Source: Wikimedia Commons [link]

Charles Babbage

English mathematician (1801, XIX century)

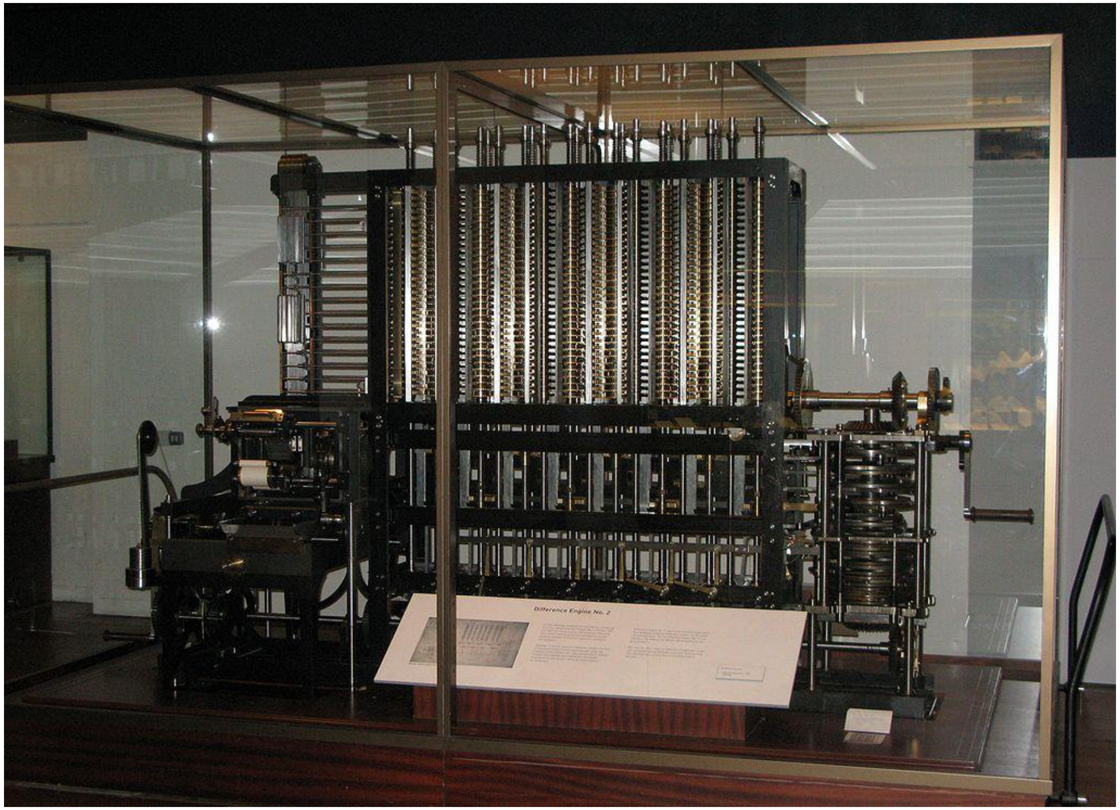
Differential Engine

Machine to calculate mathematical tables and polynomials

Analytical Engine

Antecessor of modern computers: it includes concepts such as *control unit, memory, peripherals, program,* etc.

It was designed but not built



Babbage Engine

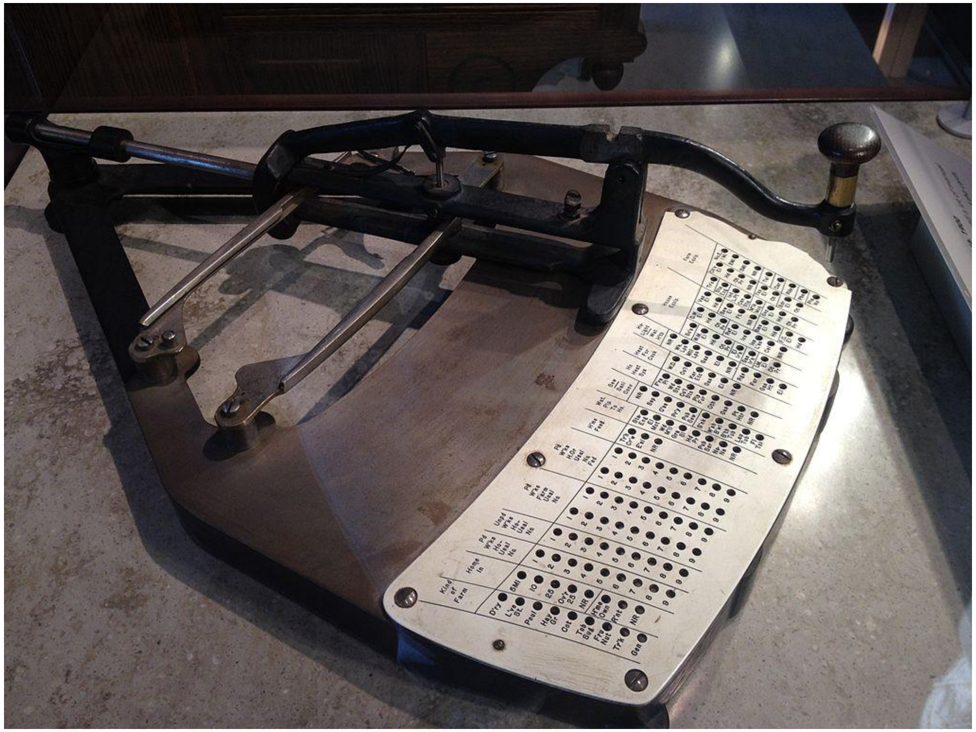
Source: Wikimedia Commons [link]

Herman Hollerith's ***Tabulating Machine***

- Census EEUU (31.4 millions inhabitants)
- Before, 12 years to process all the collected data

- The process was made automatically by using punched cards
  - Cards are classified (tabulated) and counted
  - Data is manually transcribed
- Electromechanical machine
- Time was reduced to 3 years

The tabulating machine was firstly introduced in Spain in 1925. It was installed at the Spanish Telephone Company (Compañía Telefónica Nacional de España)



Tabulating Machine

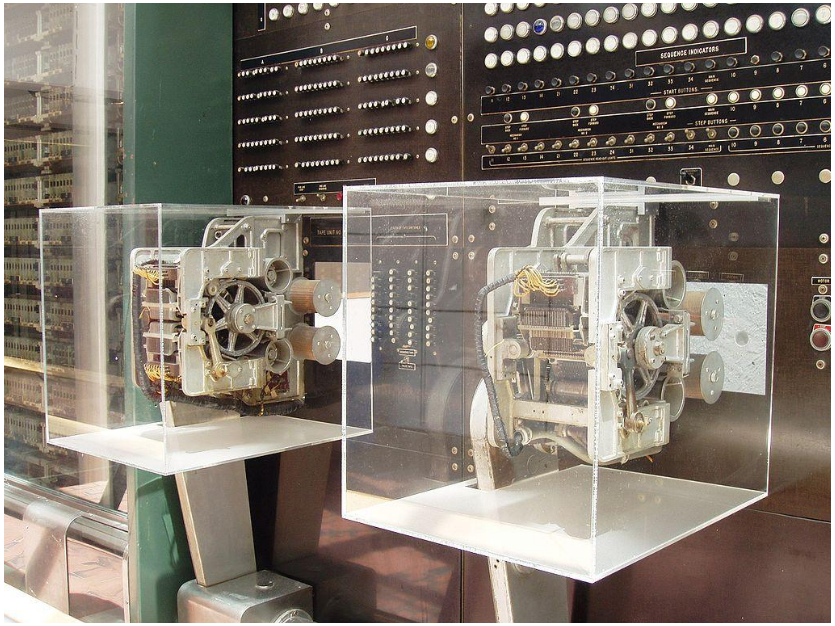
Source: Wikimedia Commons [link]

Mark I

1937 by Howard Aiken at Harvard University with IBM funds

First electromechanical computer (realization of Babbage ideas)

Composed of *relays*



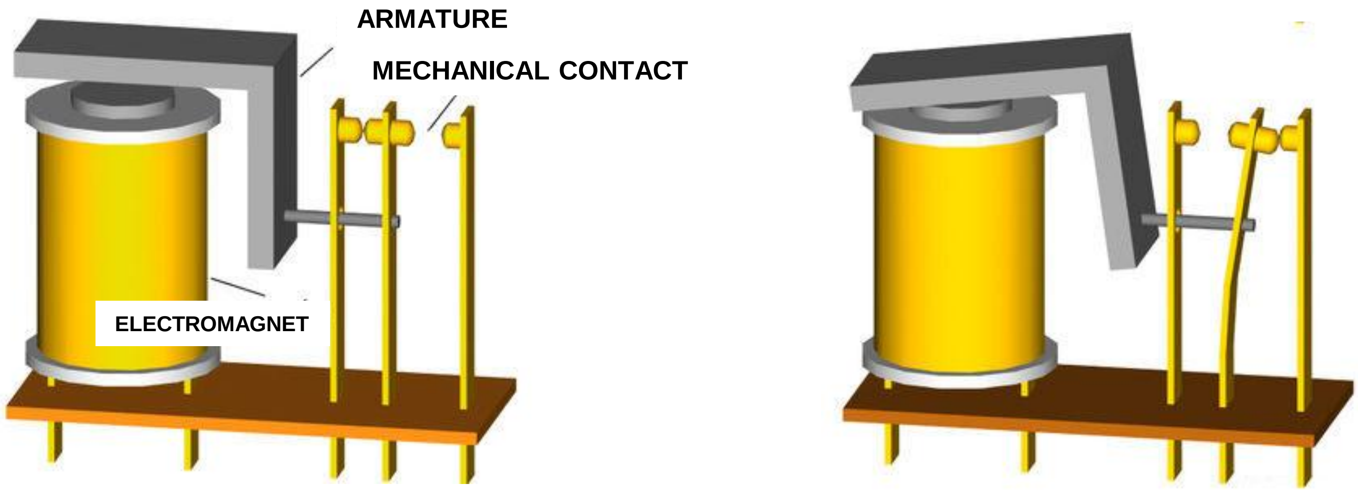
Harvard Mark I

Source: Wikemedia Commons [link]

Relay

Switch that contains a mechanical contact to open or close an electrical circuit

A small charge is applied on an electromagnet, which attracts or repulses the material of the contact



Relays
Source: Wikimedia Commons [link]

ENIAC (Electronic Numeric Integrator and Calculator)

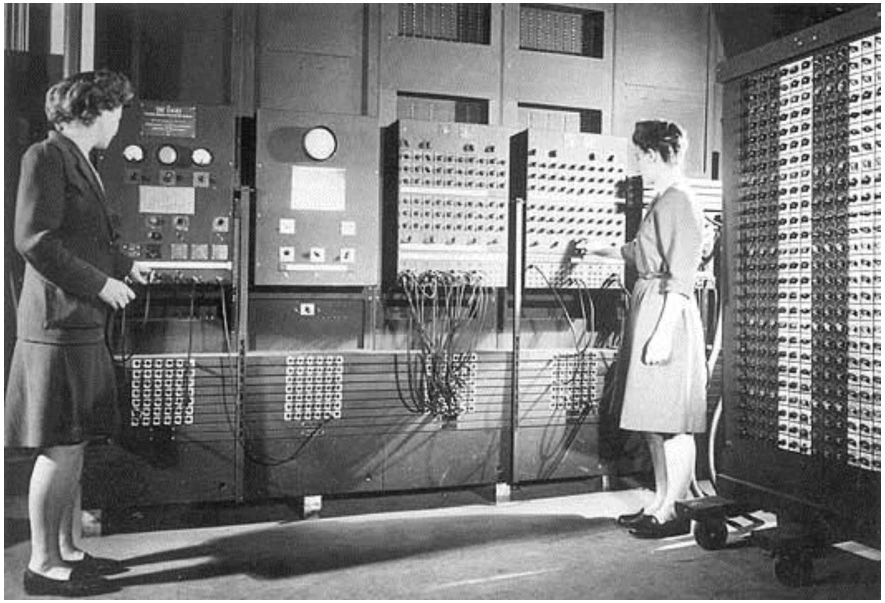
Mauchly and Eckert (1946)

First general-purpose electronic computer

Ballistics laboratory

1800 vacuum valves and 1.500 relays

Memory with 20 accumulators



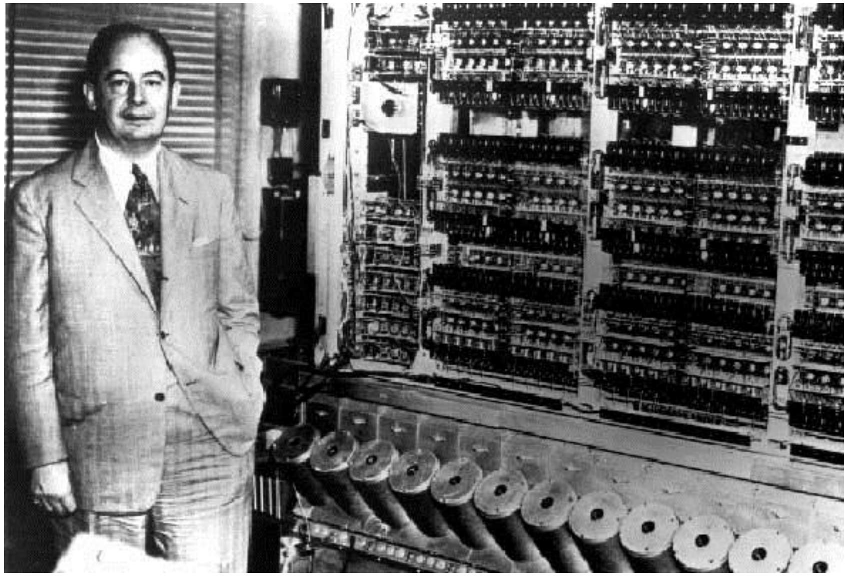
ENIAC

Source: : Wikimedia Commons [link] [video]

John von Neumann

Developed the concept of internal program: data and instructions are stored in the memory of the computer (1944)

Proposed the theory underlying modern electronic computers



John von Neumann
Source: madrimasd.org [link]

1.2. Brief history of computers

From 1950, computers increase their power and reduce their size as Electronics is developed

Usually, five generations of computers are distinguished

The transition from one generation to the next one is delimited by a significant change in the technology

First generation (1950-1960)

Vacuum tubes are used (electron tubes, thermionic valves)

Vacuum tubes require large power and produce heat

Programming in **machine language**

Data storage is performed with punched machines

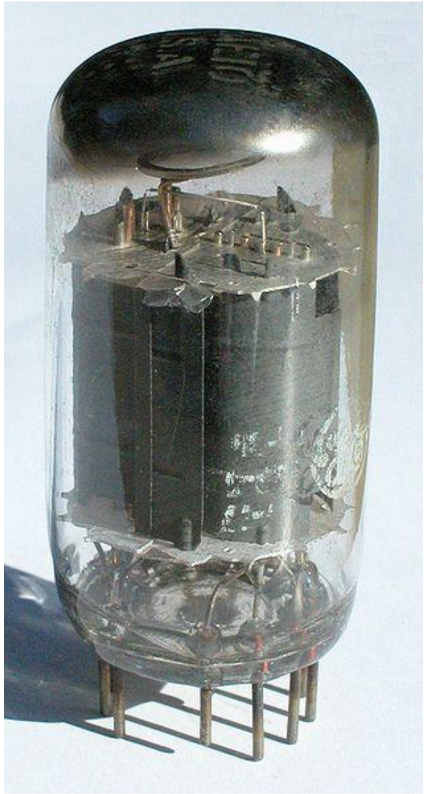
Heavy and big machines

Limited capabilities and frequent breakdowns

Military and scientific applications

Some examples:

UNIVAC I, IBM series 600 y 700



Compactron 12AE10 vacuum tube
Source: Wikipedia [link]

Second generation (1960-1966)

Transistors are used (doped silicon)

Less power, less heat

Longer life, less breakdowns

First **high-level programming languages** appear (Fortran and Cobol)

Ferrite-based (alpha iron) memories are used (magnetic core)

Computers are smaller and more reliable

Computers are fasters –hundreds of thousand operations per second

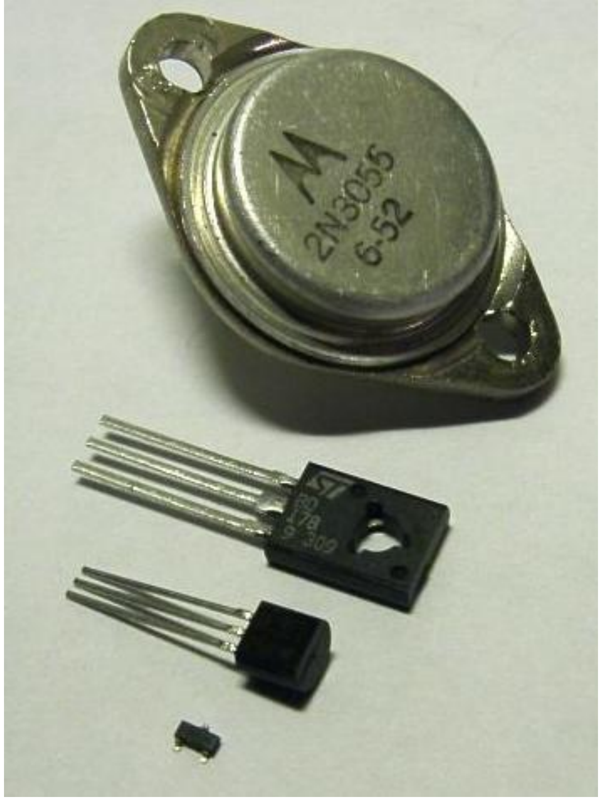
First administrative and management applications appear

Examples:

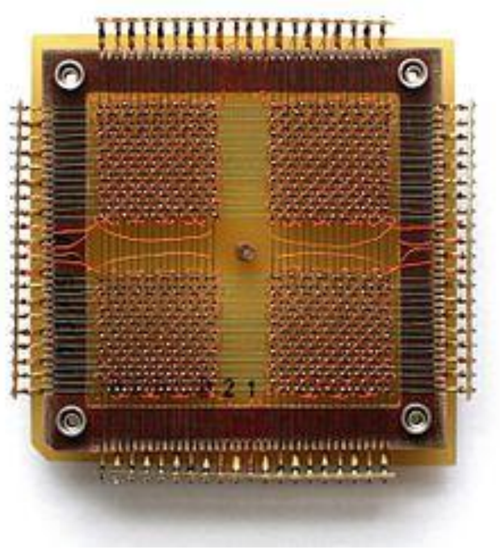
1620, 1401 y 7094 (IBM)

1107 (UNIVAC)

1.2. Brief history of computers



Transistors
Source: Wikipedia [link]



Magnetic core memory
Source: Wikipedia [link]

1.2. Brief history of computers

IBM 7090 (1959)

Source: FotosImagenes.org [link]

jgromero@inf.uc3m.es

Third generation (1966- 1971)

Integrated circuits are used (small circuits printed on silicon boards)

Very small

Size, power requirements and prize are reduced, while reliability increases

Processing speed is considerably increased (1 million of operations per second)

Great development of **operating systems**

Families of computers make it possible to use the same programs in different models

Soft disks are commercialized

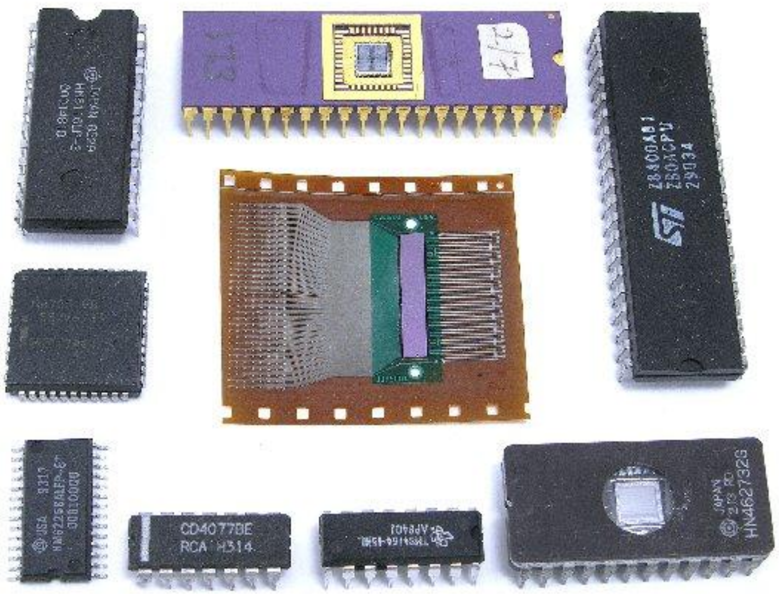
Examples

IBM 360, 370

UNIVAC 1108

Universidad Carlos III de Madrid
www.uc3m.es

1.2. Brief history of computers



Small and Medium Scale Integrated Circuits
Source: History of Electronic Devices [link]

IBM 360 Model 91
Source: Wikipedia [link]

Fourth generation (1971- 1981?)

The **microprocessor** is created

A single chip is capable of performing all the basic operations of the computer

Electronic memories are used

Great improvement in data access (speed, storage)

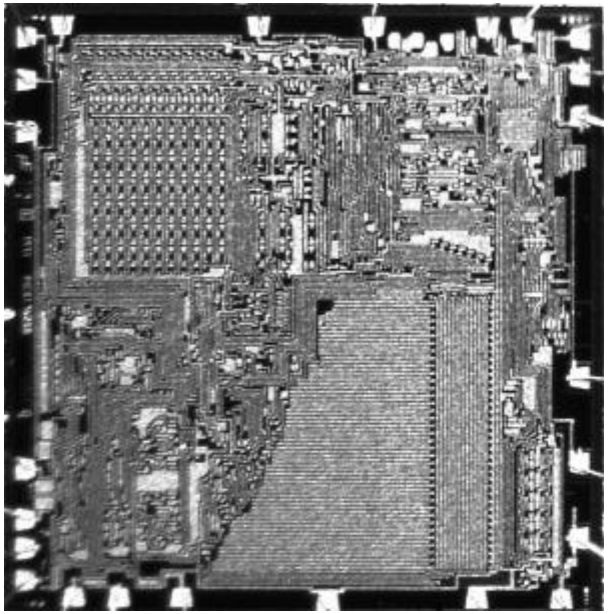
Computation power and speed continues increasing

New hardware devices are developed (peripherals, high-capacity memories)

New software tools are developed (**advanced programming languages**: C, C++)

Examples

IBM 370 (1971) IBM-PC (1981)



PICO1/GI250 microprocessor
Source: Wikipedia [link]

Examples of computer development

– Ferrite core 64Kbits = 1 meter ---> Chip 64Kbits = 1cm^2
– First IBM 370 up to 1MByte (4,674,160$ in 1970)

Fifth generation (?)

Experts do not agree to determine which is the technologic leap that lead to the new generation(s)

It will be determined in the future, with better historical perspective

Keys

Rise of personal computers (80s)

Network communications and the Internet (90s)

Ultra-portable devices and mobile phones (2000s)

Artificial Intelligence (?)

Outline

1. Computers and Computer Science
   – Computers and Computer Science
   – Brief history of computers
2. **Information representation with Computers**
3. Algorithms and programs
   – Notion of algorithm and program
   – Tools for algorithm design
4. Computer Science in the context of the Industrial Engineer

2.1. Information representation with computers

Information

Set of data structured according to some predefined order that provide knowledge about a delimited topic

Informatics

Scientific and technical knowledge that make it possible to automatically manage information by using computers

Information about a problem must be represented in a proper format to be **processed and stored** by a computer

2.1. Information representation with computers

Data, according to RAE

1. m. Antecedent necessary to achieve understanding of something or to infer the logic consequences of a fact

3. m. Inform. Information represented properly to be processed by computers

Data in Computer Science

Formal representation of facts or concepts that can be communicated or processed

Ordered data is considered information

Data **types**

Numerical (*12*, *28.5*): real (floating point) or integers

Alphabetical (*Ana*)

Alphanumerical (*23456X*, *M-6995*)

Images, sound, video, etc.

2.1. Information representation with computers

Electronic devices only have **two states**

- Relay: switch that is open or closed
- Circuit: electrical current flows or not
- Magnetic memory: polarized or not

Ultimately, the information processed by a computer is represented with 0s and 1s

- Every piece of data must be *translated* to a representation that only uses these two symbols
- **>> Binary representation**

2.1. Information representation with computers

Bit

Digits in binary notation are called bits (BIT = binary digit)

It is the smallest unit of information (0 or 1)

Byte

Set of 8 bits

It is also named character –single characters are usually encoded with one byte

jgromero@inf.uc3m.es

Integers

An integer value can be represented with binary notation

$25 = 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 10011$

An additional byte can be used to store the sign
(0: positive; 1: negative)

Representation in two-complement (more appropriate for computer processing)

Octal (base 8) and hexadecimal (base 16) are used to 'summarize' a binary representation

E.g.: These values are equivalent

Binary: **1010110001101101**

Octal: **126155** (direct conversion of groups of 3 digits, starting from the last one)

Hexadecimal: **AC6D** (direct conversion of groups of 4 digits, starting from the last one)

Real / floating point

Complex representation

The three components of the number in scientific notation are represented with different groups of bits

- mantissa/significand (significant digits)
- base/radix
- exponent

Example
0.123×10^{-4}
Mantissa: 0.123, Base: 10, Exponent: -4

Large range of numbers can be represent with the same number of bits

Standard IEEE754 (one possible representation, not the only one)

- 32 bits (extension for 64 bits – double precision)
- Base 2 (by default, no additional bits are used)
- Mantissa: 1 bit for the sign, 23 bits for the digits
- Exponent: 8 bits

Representation of real values in any digital representation always results in a loss of precision

Be careful when comparing two numbers –even if they are the same value, they might be differently encoded and considered different

Characters

Encoding: bi-univocal correspondence between a set of characters and a set of bit combinations representing numbers (by means of an *encoding table*)

Characters include numerical, alphabetical, special, and control symbols

Extended ASCII code

- American Standard Code for Information Interchange
- ASCII: 7 bits of the 8 bits of the byte are used
  - The left-most bit is not used
  - $2^7 = 128$
- Extended ASCII: 8 bits
  - $2^8 = 256$

Printable ASCII characters

Source: Wikipedia [link]

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	SP	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[	111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	|
011 1101	075	61	3D	=	101 1101	135	93	5D	]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

Representation of **analog data** (sound)

Data is sampled (*digitalized*)

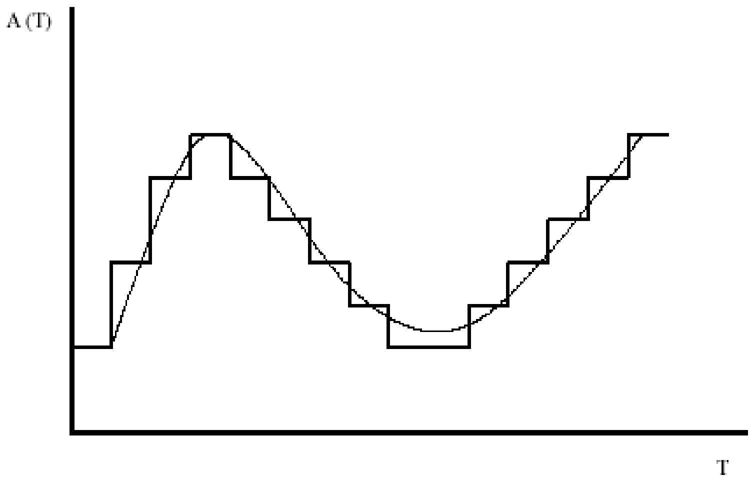
Each sample is encoded (e.g., as an integer)

Quality of the reconstructed signal (to reproduce the original sound) depends on the:

Number of samples per second

Number of bits used to encode each sample

Information is lost!



Representation of **analog data** (images)

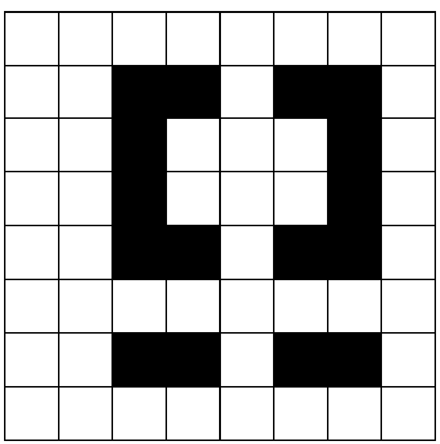
The image is sampled in a pixel matrix

Each pixel is a value representing a colour value

1-bit per pixel allow representing B/N pictures

More bits per pixel are required to represent colours

Compression techniques are used to eliminate redundant data and reduce the file size



```
0 0 0 0 0 0 0 0
0 0 1 1 0 1 1 0
0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0
0 0 1 1 0 1 1 0
0 0 0 0 0 0 0 0
0 0 1 1 0 1 1 0
0 0 0 0 0 0 0 0
```

Image quality depends on:

Resolution: number of pixels by surface unit

Number of bits per pixel (8, 16, 24, 32)

8 bits/pixel allows representing 256 different colours

40 pixels / cm
32.000 colours

8 pixels / cm
32.000 colours

256 colours

16 colours

2 colours

Outline

1. Computers and Computer Science
   – Computers and Computer Science
   – Brief history of computers
2. Information representation with Computers
3. **Algorithms and programs**
   – **Notion of algorithm and program**
   – Tools for algorithm design
4. Computer Science in the context of the Industrial Engineer

jgromero@inf.uc3m.es

3.1. Notion of algorithm and program

An algorithm is a **finite and precise** sequence of instructions to solve a problem in a finite time

It transforms input data into output results

Algorithms are used to automatically perform an operation

Examples

Algorithm to sort a list of numbers

Algorithm to allocate seats in the parliament according to vote number

Computer program

- **Set of orders** provided to a computer to perform a task
- It is the result of **translating** an algorithm into a language that the computer can understand (programming language)
- It is used to solve a problem with a computer

An **instruction** is an order issued to the computer

- A program is a set of instructions

jgromero@inf.uc3m.es

Putting everything together…

Solving a problem means to obtain output information from input information. Information is composed of data.

Computers process data by means of programs.

A program encompasses several instructions that implement an algorithm created to solve the problem.

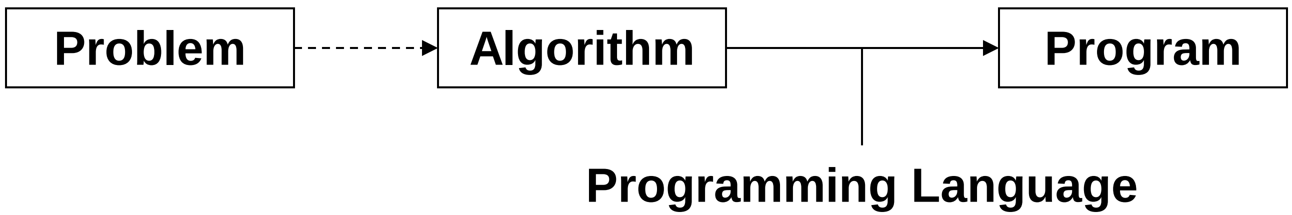
The program is stored in the computer memory and executed instruction by instruction.

As a result, the program provides the solution to the problem in the form of output data.

3.1. Notion of algorithm and program

Program: Set of orders (named instructions or sentences) written in a programming language that are provided to the computer to develop a task

- A program implements an algorithm



A program encompasses:

- Program data
  - Information processed by the program
- Expressions
  - Operations on data performed with operators
- Instructions
  - Actions of the program

jgromero@inf.uc3m.es

3.1. Notion of algorithm and program

- An algorithm must be:
  - **Precise and univocal**: concrete description
  - **Finite**: limited number of steps, limited time
  - **Correct**: it must lead to the solution
  - **Optimal**: fewest number of instructions

Algorithm to solve the problem ‘have breakfast’

```
Begin
        Sit down
        Serve coffee
        Serve sugar
        If I’m not in a hurry
                While I’m hungry Do
                        Spread butter on the toast
                        Add jam to the toast
                        Eat toast
                End While
        End If
        Drink coffee
        Stand up
End
```

Outline

1. Computers and Computer Science
   - Computers and Computer Science
   - Brief history of computers
2. Information representation with Computers
3. **Algorithms and programs**
   - Notion of algorithm and program
   - **Tools for algorithm design**
4. Computer Science in the context of the Industrial Engineer

Tools for algorithm development

Flow diagrams

Graphical representation used to detail the execution flow of an algorithm

Pseudo-code

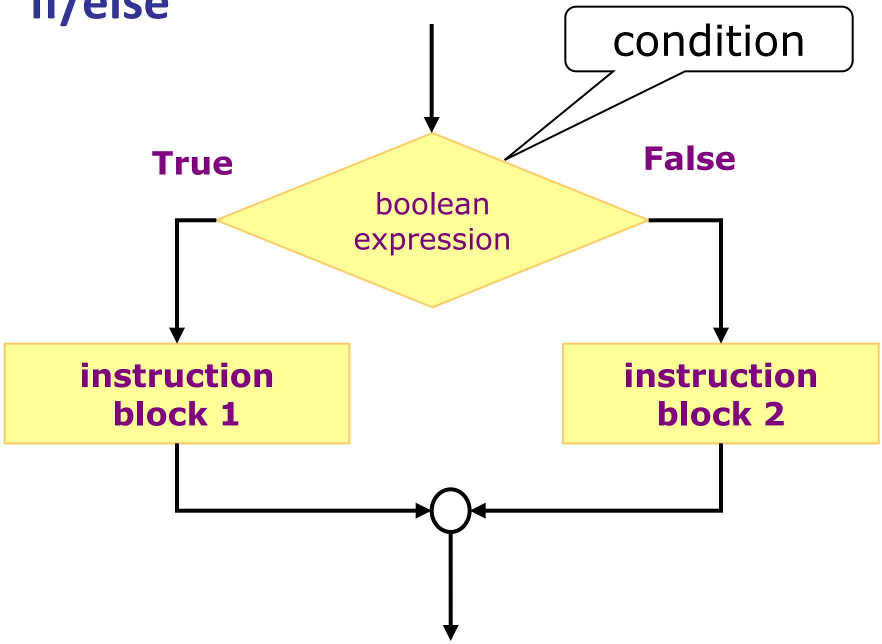
Description of the algorithm in (quasi-) natural language

Flow diagrams / Flow charts

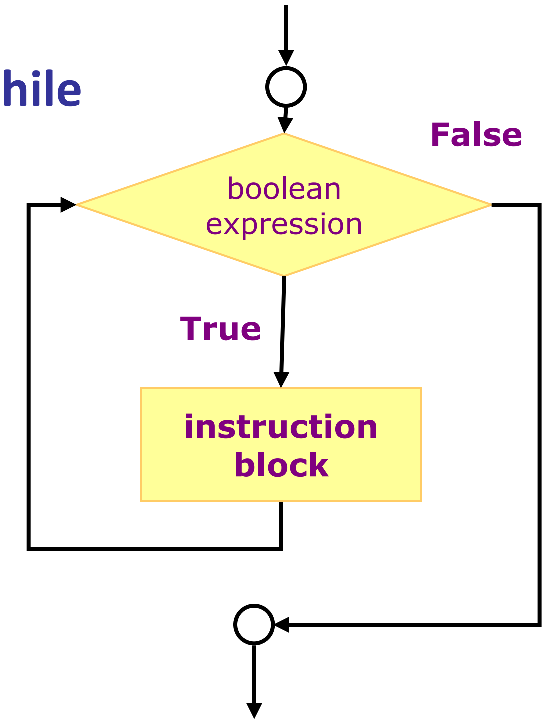
Graphical representation to detail the flow of the programs (sequence of instructions)

Standard symbols

if/else

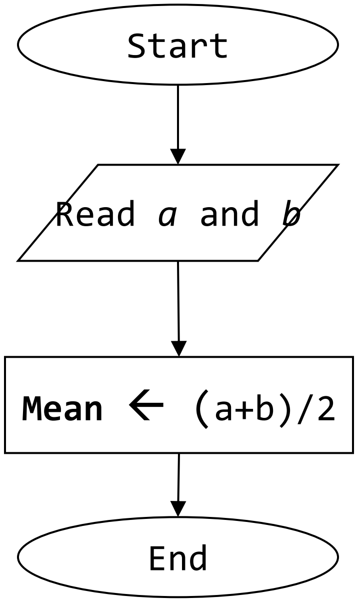


while

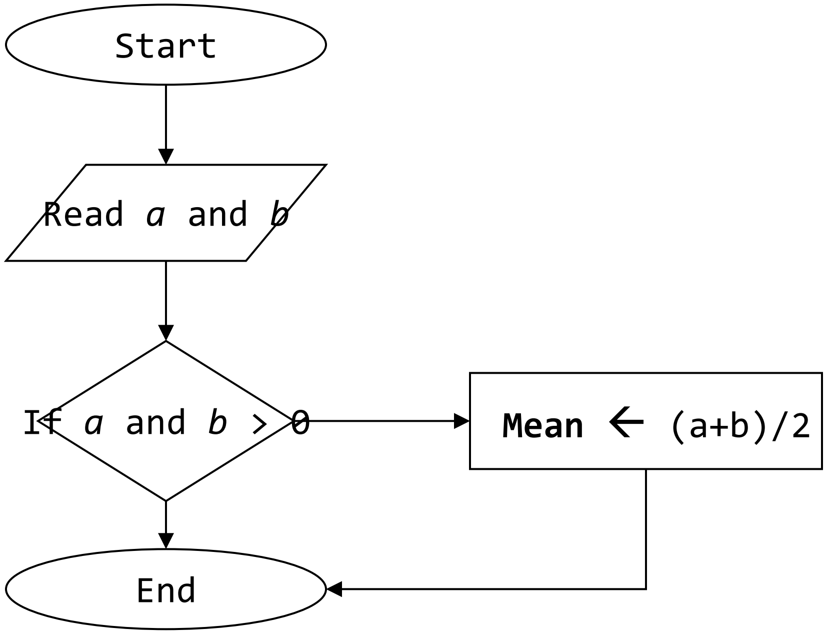


Flow diagrams

Mean of two values



Mean of two positive values



jgromero@inf.uc3m.es

Pseudo-code

The procedure is described with words very similar to the natural language

Intermediate stage between the algorithm and the implementation in a programming language

Pseudo-code programs include

- Name of the program
- Environment: elements that are used by the program (variables, constants) and type
- Instructions

```
Program1

ENV
    variables

START
    instruc_1
    instruc_n
END
```

```
Mean

ENV
    a,b,mean (real numbers)
START
    read a and b
    mean ← (a+b)/2
    print mean
END
```

Pseudo-code

Conditional instructions

```
If (expression) Then
        I1,I2,…
End If
```

```
If (expression) Then
        I1,I2,…
Else
        S1,S2,…
End If
```

Pseudo-code

Loop instructions

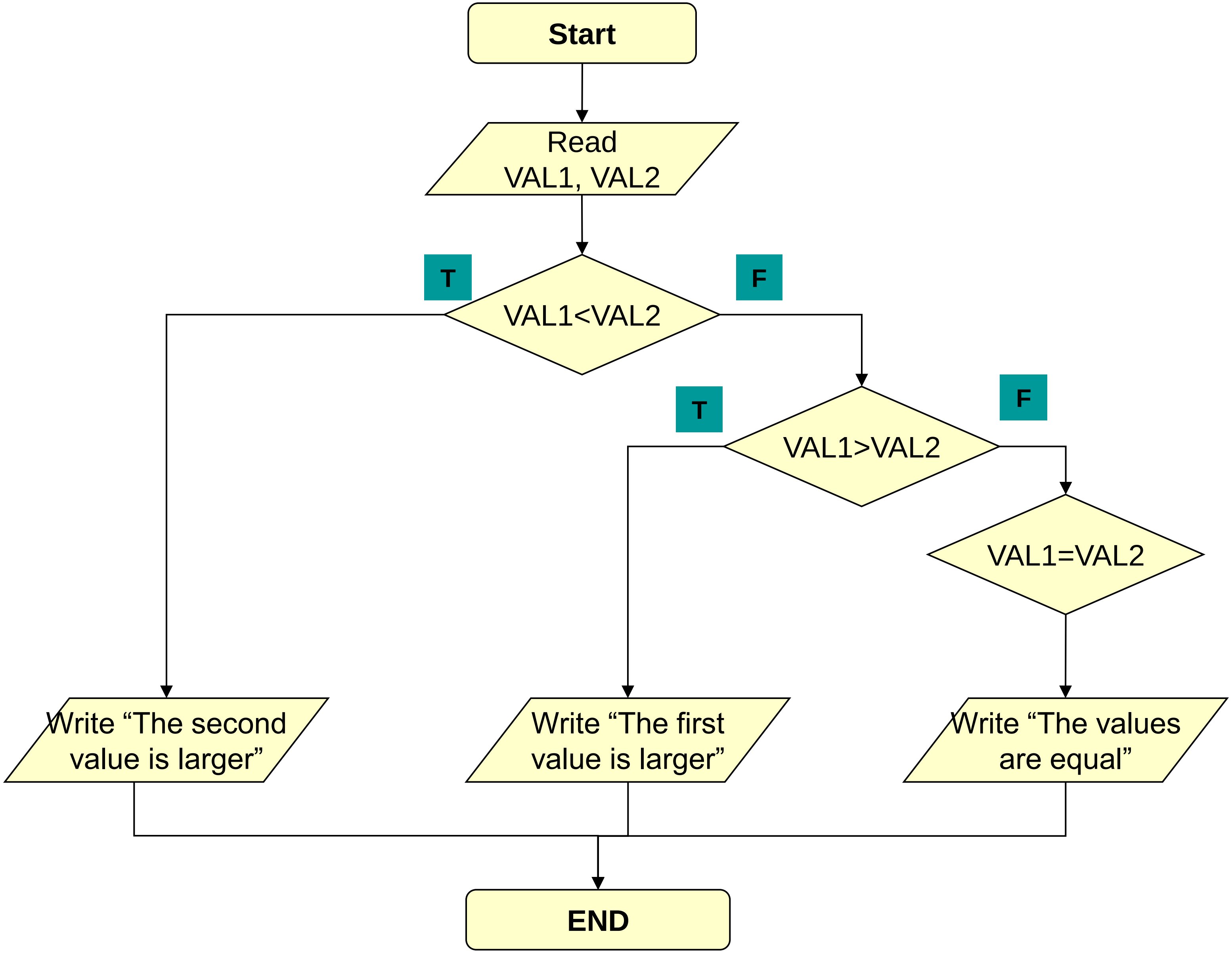
- Repeat an instruction (or a block of instructions)
- Examples:
  - Algorithm to print 100 times ‘Hello’
  - Algorithm to add 100 numbers read from the keyboard
- Two types: `for` and `while`
- May lead to infinite loops – programs that never end

```
While (expression) do
        I1,I2,…
End While
```

```
For variable starting from init value until end value in steps of value do
        I1,I2,….
End For
```

Develop an algorithm that, given two numbers, find out the largest of them

```
Programa Largest
ENV
        VAL1: First value, real
        VAL2: Second value, real
START
        Read VAL1
        Read VAL2
        If ( VAL1>VAL2) Then
                write  'The first value is larger'
        Else
                If( VAL1<VAL2 ) Then
                        write 'The second value is larger'
                Else
                        write 'The values are equal'
                End If
        End If
END
```


Start
Read
VAL1, VAL2
T
VAL1<VAL2
F
T
VAL1>VAL2
F
VAL1=VAL2
Write “The second value is larger”
Write “The first value is larger”
Write “The values are equal”
END

Exercises

Develop a proper algorithm for the following problems. Express each algorithm with a flow diagram and pseudo-code

1. Determine if the user is over 18 years
2. Calculation of the solution of a second grade equation
3. Guess the number game
   - A value is chosen by the computer
   - A value is read from the keyboard
     - If the typed value is larger than the value to find, the following message is printed: “Your guess is larger than the value”
     - If the typed value is less than the value to find, the following message is printed: “Your guess is less than the value”. The game continues (the user introduces a new value).
     - If the typed value is equal to the value to find, the following message is printed: “You find the number!”. The game ends.

Outline

1. Computers and Computer Science
   - Computers and Computer Science
   - Brief history of computers
2. Information representation with Computers
3. Algorithms and programs
   - Notion of algorithm and program
   - Tools for algorithm design
4. **Computer Science in the context of the Industrial Engineer**

4. Computer Science in the context of the Industrial Engineer

Informatics is present in almost every daily task

In **Industrial Engineering**

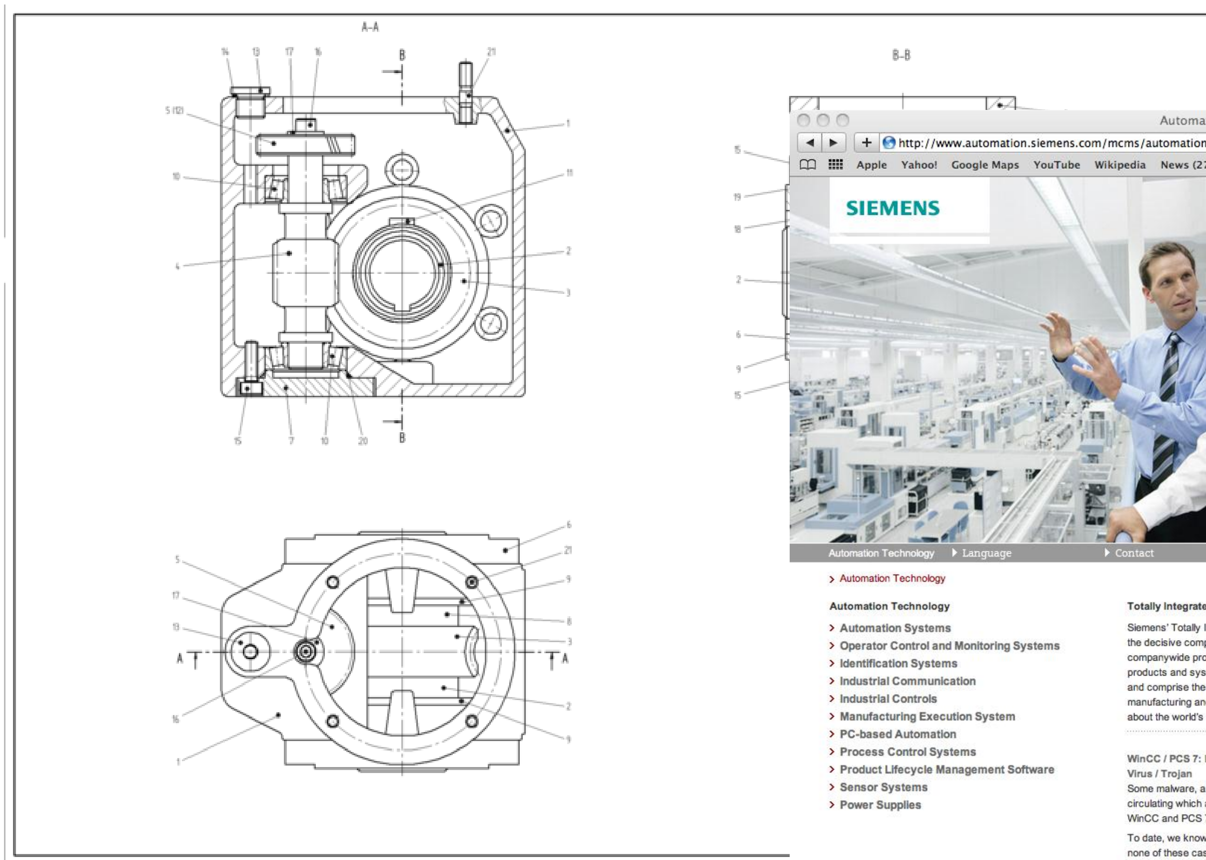
- Industries are more and more **automatized**
  - Industrial processes are performed by robots or automatic tools
- The Industrial Engineer *designs, develops, test, and implements* solutions to industrial problems
- These solutions **necessarily involve computer systems**
- The Engineer should know
  - The functioning of systems based on computational processors
  - Programming techniques
    - > Mainly in Automation and Control applications
  - Fundamentals of the functioning of software applications

Some applications

- Computer-assisted design, production, and test (CAD/CAM tools)
- Industrial Informatics
  - Computers are used to control individual machines or large engineered systems (e.g., petrochemical plant)
- Robotics
- Geographic Information Systems (GIS)

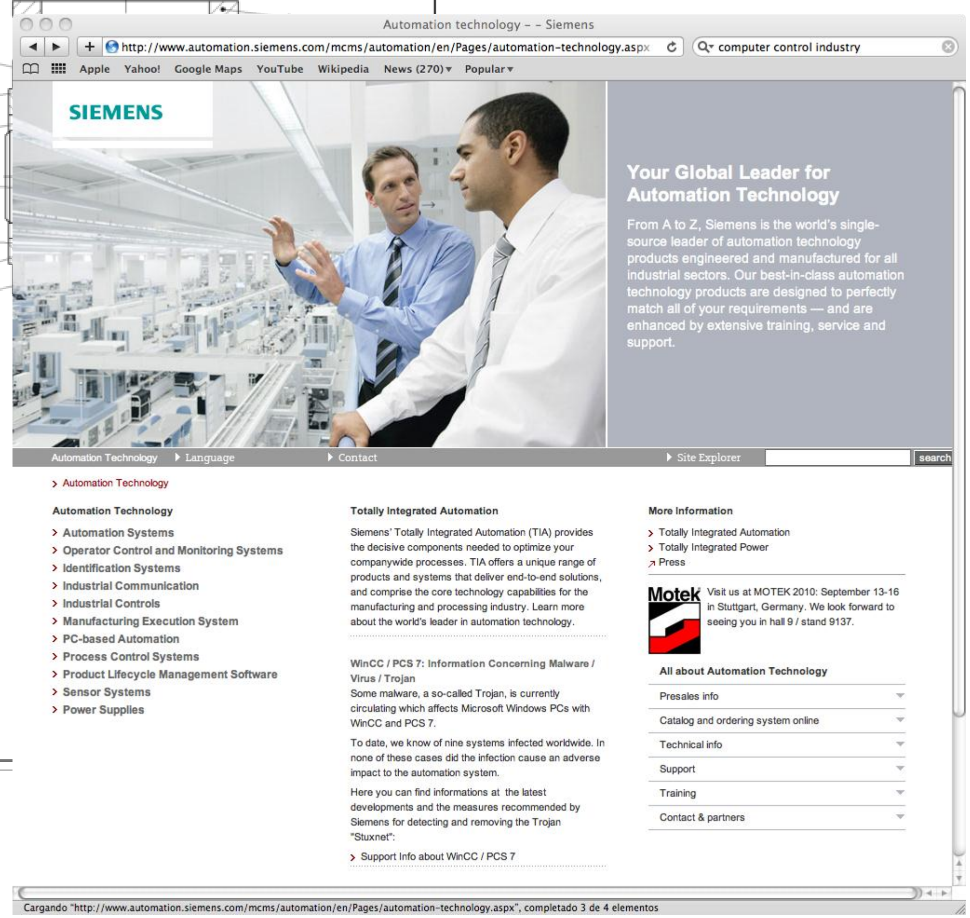
Universidad Carlos III de Madrid
www.uc3m.es

4. Computer Science in the context of the Industrial Engineer



CAD modeling

Source: Wikipedia [link]



Automation tools

Source: Siemens [link]

Outline

1. Computers and Computer Science
   – Computers and Computer Science
   – Brief history of computers
2. Information representation with Computers
3. Algorithms and programs
   – Notion of algorithm and program
   – Tools for algorithm design
4. Computer Science in the context of the Industrial Engineer