METU Computer Engineering

# Ceng 111 – Fall 2021 Week 1b

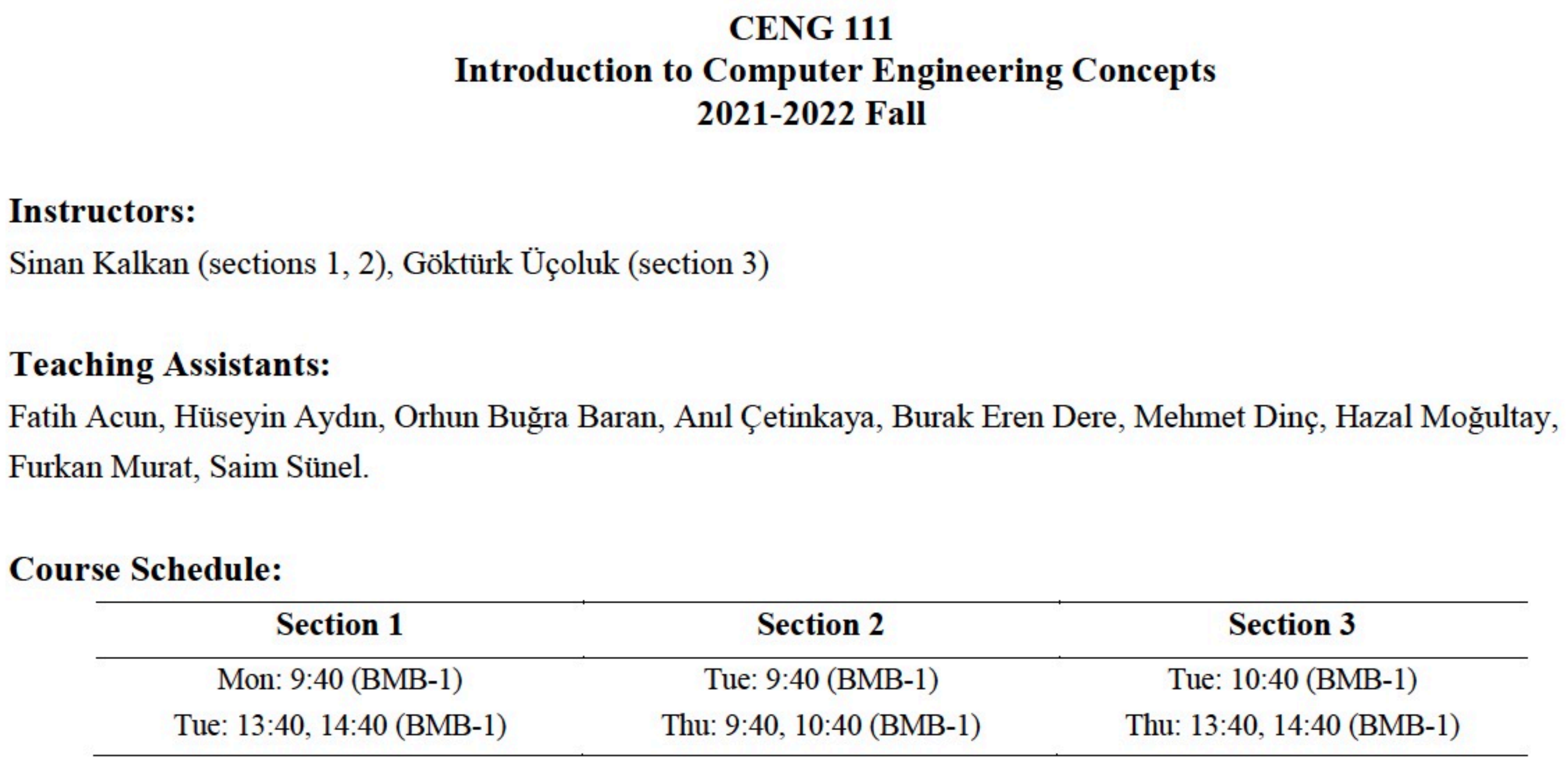
Introduction to Computing

**Credit**: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.

Previously on CENG111!

## Syllabus

https://user.ceng.metu.edu.tr/~skalkan/ceng111_syllabus.pdf



**CENG 111**
**Introduction to Computer Engineering Concepts**
**2021-2022 Fall**

**Instructors:**

Sinan Kalkan (sections 1, 2), Göktürk Üçoluk (section 3)

**Teaching Assistants:**

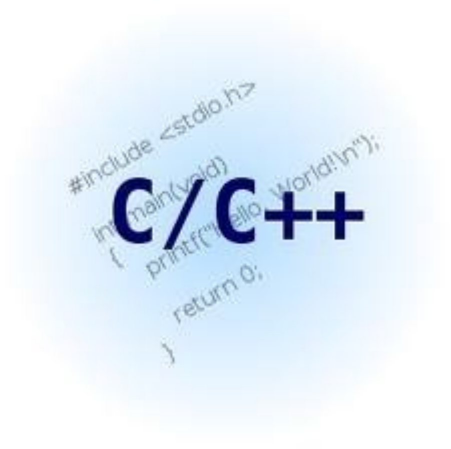
Fatih Acun, Hüseyin Aydın, Orhun Buğra Baran, Anıl Çetinkaya, Burak Eren Dere, Mehmet Dinç, Hazal Moğultay, Furkan Murat, Saim Sünel.

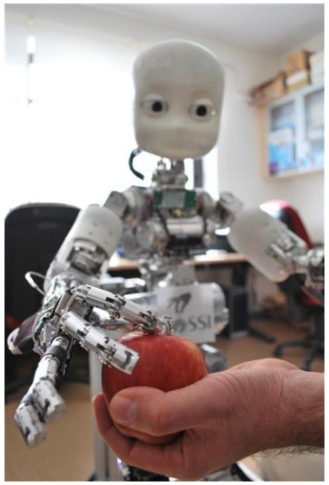
**Course Schedule:**

| **Section 1** | **Section 2** | **Section 3** |
|---|---|---|
| Mon: 9:40 (BMB-1)<br>Tue: 13:40, 14:40 (BMB-1) | Tue: 9:40 (BMB-1)<br>Thu: 9:40, 10:40 (BMB-1) | Tue: 10:40 (BMB-1)<br>Thu: 13:40, 14:40 (BMB-1) |

## Today

- Introduction to Computer Science
  - What CS is and what it is not
  - Computing, computation, algorithm





## WHAT IS COMPUTER SCIENCE?

## What is Computer Science?

- *"Computer science is the study of computers"* WRONG!
- Incomplete – theoretical work began (1920-1940) before computers
- CS became an independent field of study late 1950's, early 1960's
- Theoretical CS – relies on formal models rather than "real" machines
- CS "is no more about computers than astronomy is about telescopes"

**From "Invitation to Computer Science"**

## What is Computer Science?

WRONG!

- *"Computer science is the study of how to write computer programs"*

- Programming is important, but it is just a tool for studying new ideas, representing information or testing the solution to a problem.
- A Program is a means to an end, not the end itself.

**From "Invitation to Computer Science"**

## What is Computer Science?

- *"Computer science is the study of the uses and applications of computers and software"*

WRONG!

- E.g. word processors, databases, spreadsheets, etc.
- Many people USE software, but the Computer Scientist is responsible for <u>specifying, designing, building and testing</u> software packages and the systems on which they run.

**From "Invitation to Computer Science"**

## What is Computer Science?

- All of the following concepts are incomplete and do not capture the richness and diversity of this exciting field:
  - computers,
  - programming languages,
  - software applications, and uses.

From "Invitation to Computer Science"

## Computer Science is the study of algorithms (= methods)

including:

1. their formal and mathematical properties
2. their hardware realizations
3. their linguistic realizations
4. their applications

**From "Invitation to Computer Science"**

## Computer Science is the study of Algorithms

Including:

1. their formal and mathematical properties
   - studying the behavior of algorithms to see that they are correct and efficient

**From "Invitation to Computer Science"**

## Computer Science is the study of Algorithms

Including:

2. their hardware realizations
   - designing and building computer systems to execute the algorithms

**From "Invitation to Computer Science"**

## Computer Science is the study of Algorithms

Including:

3. their linguistic realizations
   - designing programming languages and translating the algorithms into these languages so that they can be executed by the hardware

From "Invitation to Computer Science"

# Computer Science is the study of Algorithms

Including:

4. their applications
   - identifying important problems and designing correct and efficient software packages to solve them.

From "Invitation to Computer Science"

## Measure the height of a tall building with a barometer

The Barometer Problem
NOW, CLASS-- DO ANY OF YOU KNOW HOW TO MEASURE THE HEIGHT OF A TALL BUILDING USING A BAROMETER?

- What would be your answer?
- One student answered:
  - "I would tie the barometer to a rope, hang it down from the top of the building to the bottom and measure the length of the rope!"
  - Of course, the instructor rejects the answer since it doesn't include any "physics"

Check the following for two different versions of the `legend':
http://www.snopes.com/college/exam/barometer.asp

METU Computer Engineering

## Measure the height of a tall building with a barometer (cont’d)

- <u>There are other ways:</u>
  1. Drop the barometer from the top, and measure the time it takes to reach the ground.
  2. Make a pendulum and time its period wrt. the top and the bottom of the building.
  3. Walk down the stairs marking "barometer units" on the wall.
  4. Measure its shadow and the buildings shadow. Workout the height of the building from barometer’s height.

# Analyze/Compare Algorithms = Pros and Cons of Algorithms =

What are the disadvantages of these?

1. Drop the barometer from the top, and measure the time it takes to reach the ground.
2. Make a pendulum and time its period wrt. the top and the bottom of the building.
3. Walk down the stairs marking "barometer units" on the wall.
4. Measure its shadow and the buildings shadow. Workout the height of the building from barometer's height.

**Cons:**

1. You lose the barometer; it breaks.
2. You need a loooong rope and how long that rope is going to be depends on the answer to the question.
3. It takes too long. It is too tiring.
4. What if there is no sun?

## Where does the word ‘algorithm’ come from?

- From a Persian mathematician, astronomer and geographer: Mohammed ibn-Musa al-Khwarizmi
- “Algorithmi” is the latin form of his name
- He contributed to science by
  - Decimal positional number system (e.g., 32 = $10^1$x3 + $10^0$x2)
  - Presented the first systematic solutions to linear and quadratic equations
- In fact, the word “Algebra” comes from one of his operators (al-jabr: subtracting a number from both sides of an equation) for solving equations

Mohammed ibn-Musa al-Khwarizmi (780-850)

**Source**: Wikipedia

# Where does the word 'algorithm' come from? (cont'd)

- al-Khwarizmi reduced equations to one of the following six forms by using al-jabr (in Arabic: restoring, completion):
  - squares equal roots ($ax^2 = bx$)
  - squares equal number ($ax^2 = c$)
  - roots equal number ($bx = c$)
  - squares and roots equal number ($ax^2 + bx = c$)
  - squares and number equal roots ($ax^2 + c = bx$)
  - roots and number equal squares ($bx + c = ax^2$)
- For example, $x^2 = 40x - 4x^2$ is reduced to $5x^2 = 40x$. From this reduced form, it is easily deducable that the variable is either 0 or 8.

كتاب الخوارزمي

[illegible]

al-Khwarizmi's Compendious Book on Calculation by Completion and Balancing

**Source**: Wikipedia

## What does 'algorithm' mean?

- "A procedure or formula for solving a problem"
- "A set of instructions to be followed to solve a problem"
- "an effective method expressed as a finite list of well-defined instructions for calculating a function"
- "step-by-step procedure for calculations"

## A formal definition of algorithm

- "Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, will proceed through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state."

## What is an algorithm?

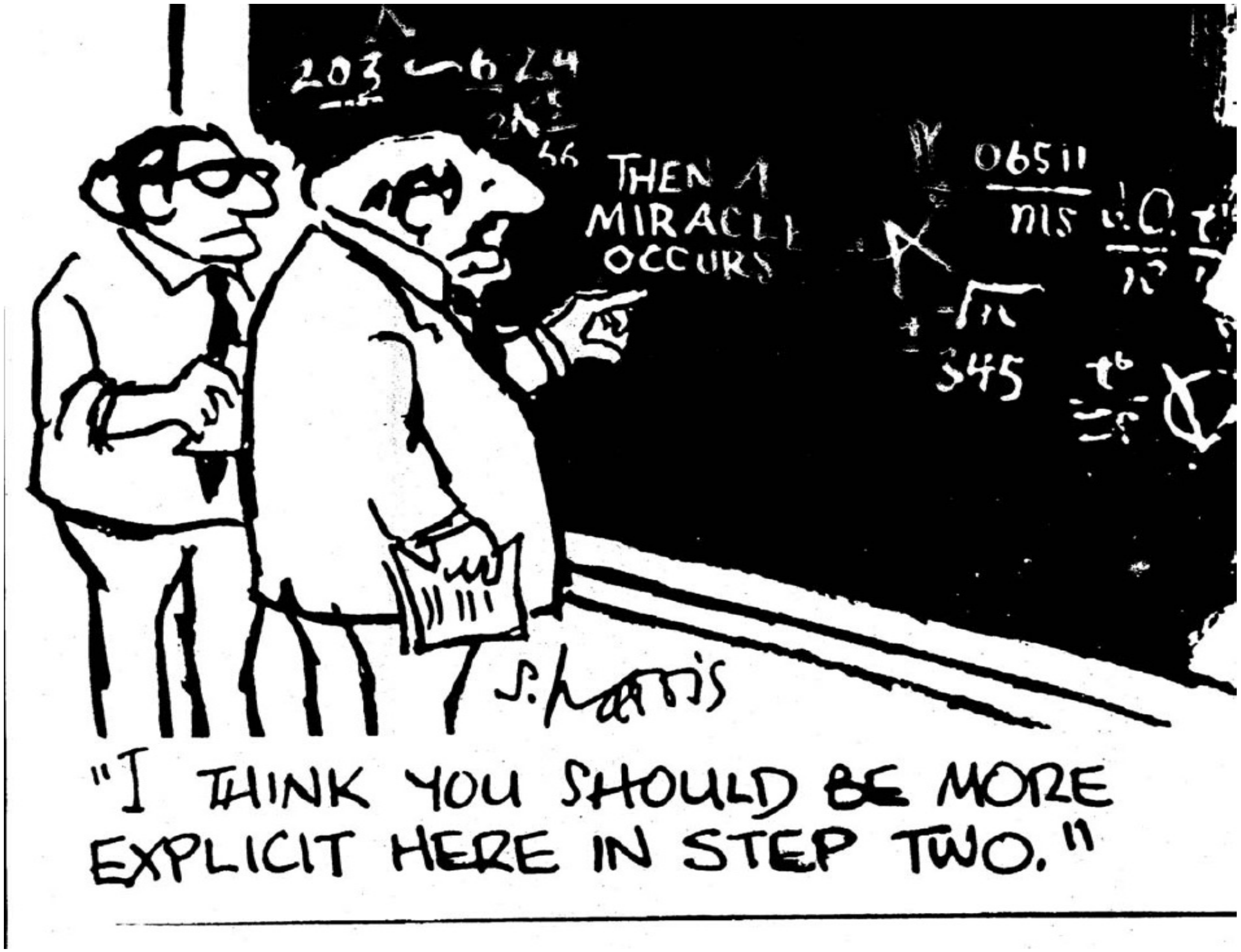
- An algorithm is a list that looks like
  - STEP 1: Do something
  - STEP 2: Do something
  - STEP 3: Do something
  - . .
  - . .
  - . .
  - STEP N: Stop, you are finished

**From “Invitation to Computer Science”**

## Valid Operations in Algorithms

- **Sequential** – simple well-defined task, usually declarative sentence.
- **Conditional**- "ask a question and select the next operation on the basis of the answer to the question – usually an "if-then" or "if then else"
- **Iterative**- "looping" instructions – repeat a set of instructions

**From "Invitation to Computer Science"**



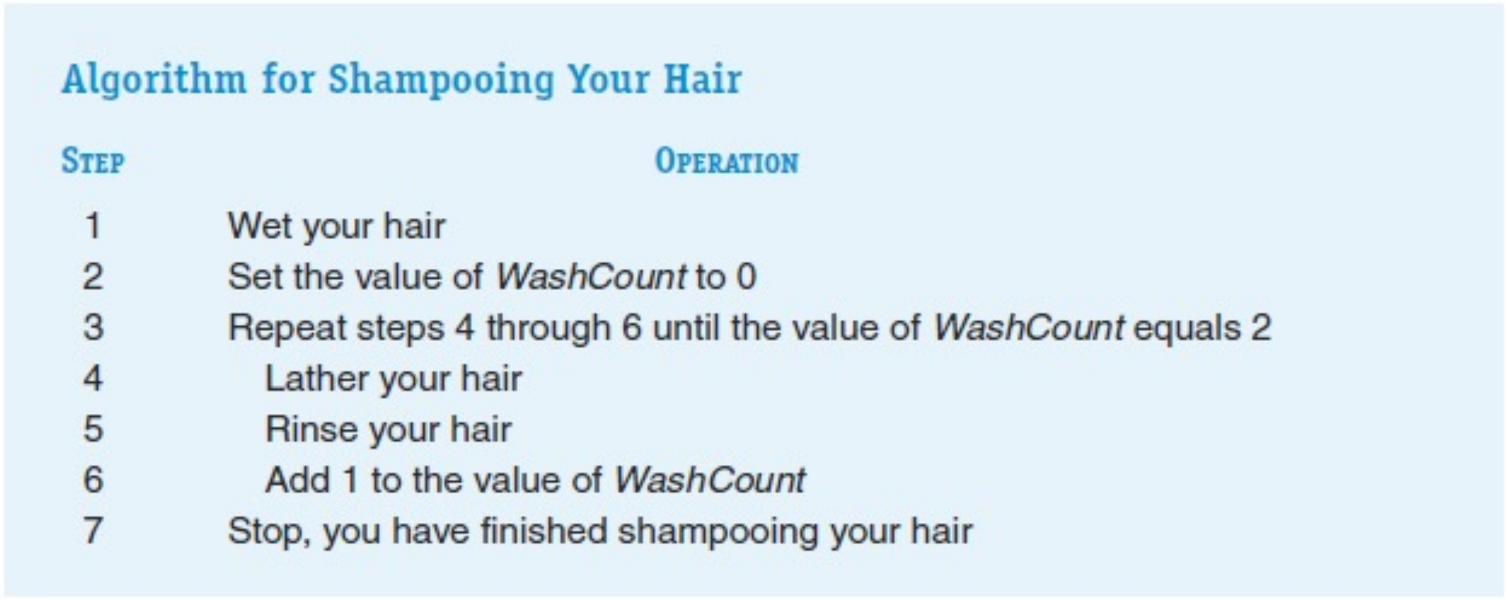
From "Invitation to Computer Science"

## Algorithms

- We use them all the time.
- Can you give examples?
  - Following directions
  - Recording a DVD
  - Adding two numbers
  - Finding Greatest Common Divisor
  - …

**From "Invitation to Computer Science"**

## An example algorithm from our daily lives



**Algorithm for Shampooing Your Hair**

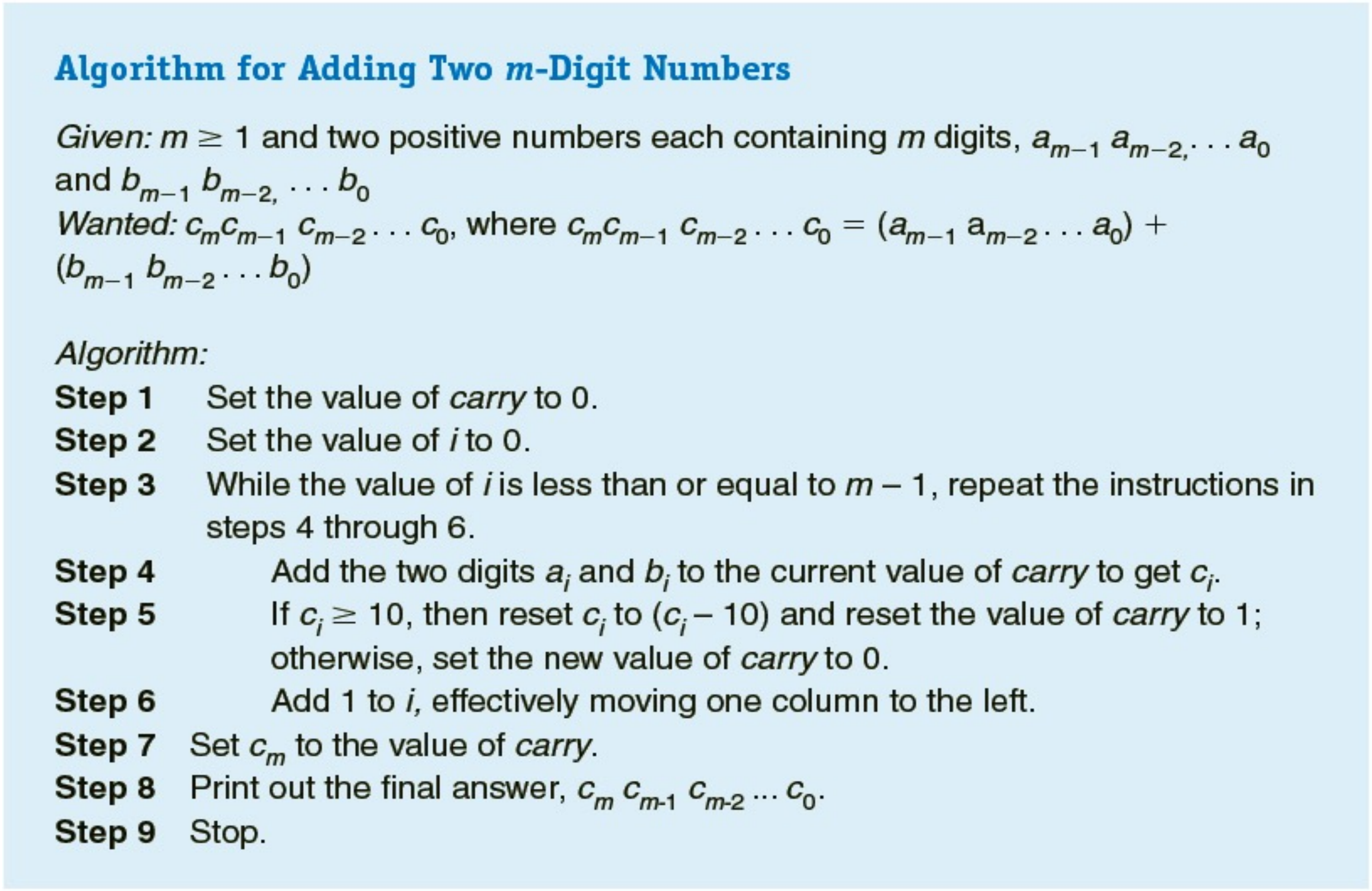
| Step | Operation |
|---|---|
| 1 | Wet your hair |
| 2 | Set the value of *WashCount* to 0 |
| 3 | Repeat steps 4 through 6 until the value of *WashCount* equals 2 |
| 4 |     Lather your hair |
| 5 |     Rinse your hair |
| 6 |     Add 1 to the value of *WashCount* |
| 7 | Stop, you have finished shampooing your hair |

**From “Invitation to Computer Science”**

THIS "EXACT INSTRUCTIONS CHALLENGE" IS SO HILARIOUS

https://www.youtube.com/watch?v=Ct-lOOUqmyY

# An example algorithm



**Algorithm for Adding Two *m*-Digit Numbers**

*Given:* $m \geq 1$ and two positive numbers each containing $m$ digits, $a_{m-1}\ a_{m-2,} \ldots a_0$ and $b_{m-1}\ b_{m-2,} \ldots b_0$

*Wanted:* $c_m c_{m-1}\ c_{m-2} \ldots c_0$, where $c_m c_{m-1}\ c_{m-2} \ldots c_0 = (a_{m-1}\ a_{m-2} \ldots a_0) + (b_{m-1}\ b_{m-2} \ldots b_0)$

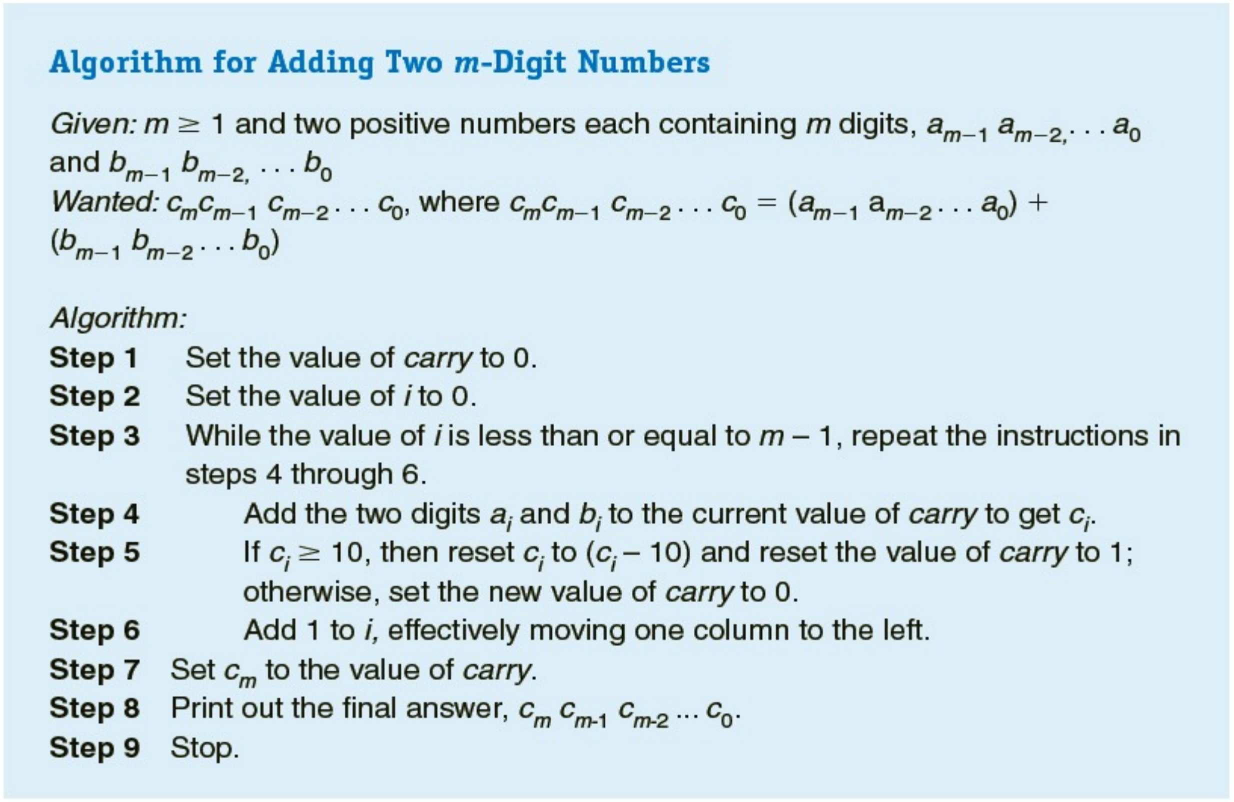
*Algorithm:*

```
Step 1   Set the value of carry to 0.
Step 2   Set the value of i to 0.
Step 3   While the value of i is less than or equal to $m - 1$, repeat the instructions in
         steps 4 through 6.
Step 4       Add the two digits $a_i$ and $b_i$ to the current value of carry to get $c_i$.
Step 5       If $c_i \geq 10$, then reset $c_i$ to $(c_i - 10)$ and reset the value of carry to 1;
             otherwise, set the new value of carry to 0.
Step 6       Add 1 to i, effectively moving one column to the left.
Step 7   Set $c_m$ to the value of carry.
Step 8   Print out the final answer, $c_m\ c_{m-1}\ c_{m-2} \ldots c_0$.
Step 9   Stop.
```

**From "Invitation to Computer Science"**

# How to represent algorithms

- Pseudo-code



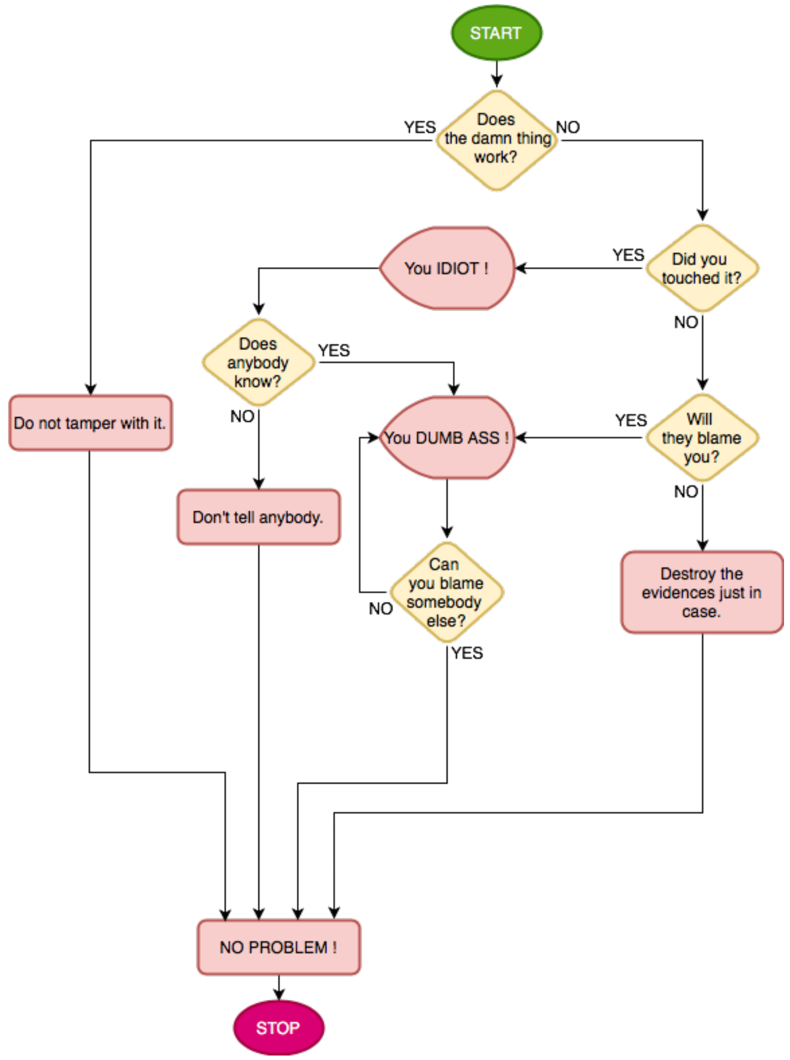
**Algorithm for Adding Two $m$-Digit Numbers**

*Given:* $m \geq 1$ and two positive numbers each containing $m$ digits, $a_{m-1}\ a_{m-2}, \ldots a_0$ and $b_{m-1}\ b_{m-2}, \ldots b_0$
*Wanted:* $c_m c_{m-1}\ c_{m-2} \ldots c_0$, where $c_m c_{m-1}\ c_{m-2} \ldots c_0 = (a_{m-1}\ a_{m-2} \ldots a_0) + (b_{m-1}\ b_{m-2} \ldots b_0)$

```
Algorithm:
Step 1  Set the value of carry to 0.
Step 2  Set the value of i to 0.
Step 3  While the value of i is less than or equal to $m - 1$, repeat the instructions in steps 4 through 6.
Step 4      Add the two digits $a_i$ and $b_i$ to the current value of carry to get $c_i$.
Step 5      If $c_i \geq 10$, then reset $c_i$ to $(c_i - 10)$ and reset the value of carry to 1;
            otherwise, set the new value of carry to 0.
Step 6      Add 1 to i, effectively moving one column to the left.
Step 7  Set $c_m$ to the value of carry.
Step 8  Print out the final answer, $c_m\ c_{m-1}\ c_{m-2} \ldots c_0$.
Step 9  Stop.
```

# How to represent algorithms

- Flow-charts



## Why are algorithms important?

- If we can specify an algorithm to solve a problem then we can automate its solution.
- No algorithm => No software => No automation!

**From "Invitation to Computer Science"**

## Can we find algorithms to all problems?

NO!

- There are problems which have no generalized solutions – unsolvable or intractable
- Some with an algorithm would take so long to execute that the algorithm is useless
- Some problems we have not yet discovered an algorithm for

**From "Invitation to Computer Science"**

## A formal definition of algorithm

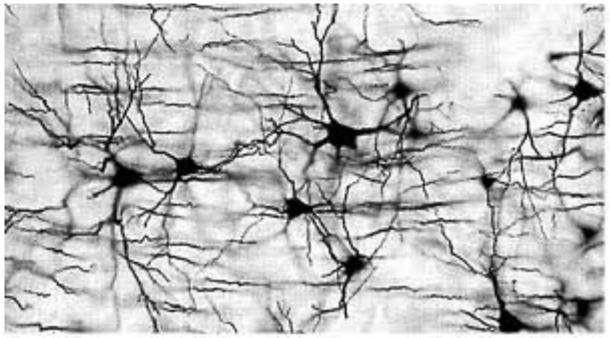
- “Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, will proceed through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state.”

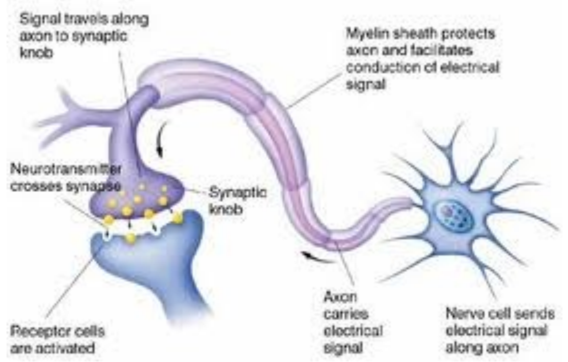
# "Computation"

- Digital vs. analog computation
- Sequential vs. parallel computation
- Batch vs. interactive computation
- Evolutionary, molecular, quantum computation
- "Physical computation" / "Digital Physics"
  - 'The whole universe is itself a computation'

## Computation in our brain

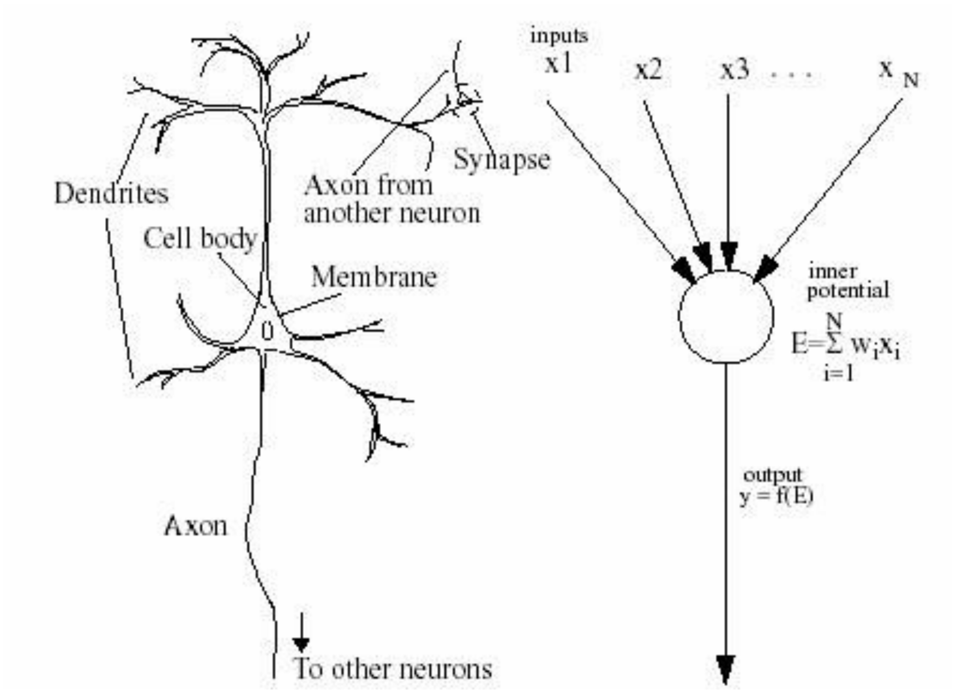
- Highly-connected network of neurons.
- How many neurons?
  - Approx. $10^{11}$ neurons and $10^{14}$ synapses.
- How do they transmit information?
  - Using nothing else than charged molecules.



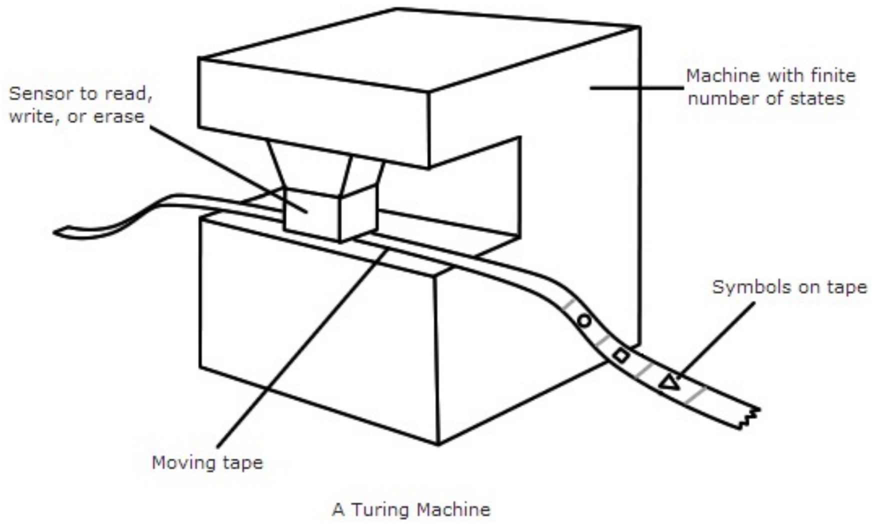

Signal travels along axon to synaptic knob
Myelin sheath protects axon and facilitates conduction of electrical signal
Neurotransmitter crosses synapse
Synaptic knob
Axon carries electrical signal
Nerve cell sends electrical signal along axon
Receptor cells are activated

## Computation in our brain (cont'd)

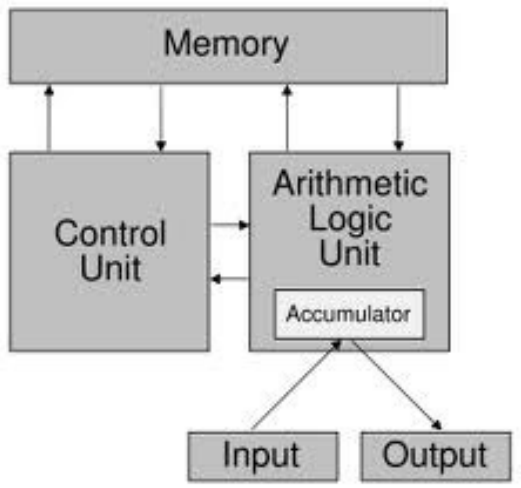
- Each neuron gets input and produces an output using an "activation function"



- Some of ours' is smaller but they have essentially the same computational mechanisms! ☺


Sensor to read, write, or erase
Machine with finite number of states
Symbols on tape
Moving tape
A Turing Machine

Turing Machine



Von Neumann Architecture

## DIGITAL COMPUTATION

## BUT FIRST SOME HISTORICAL OVERVIEW

# The Early Period: Up to 1940

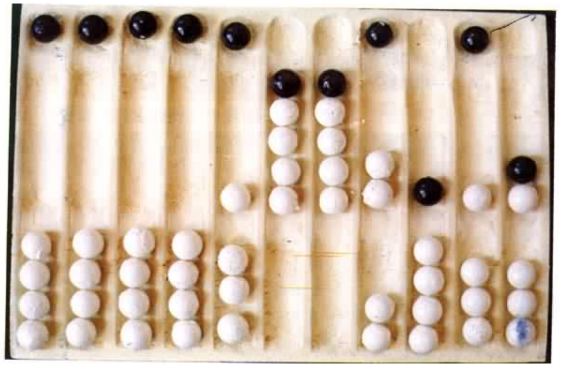
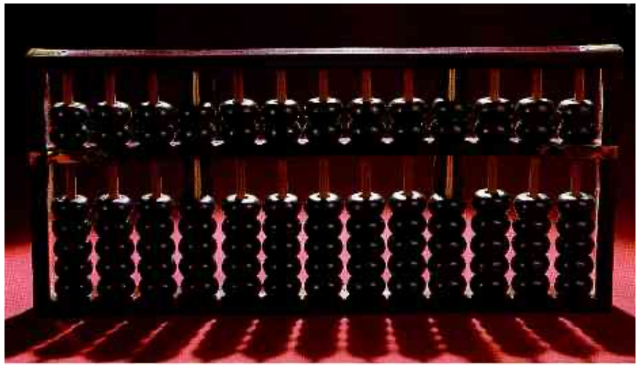
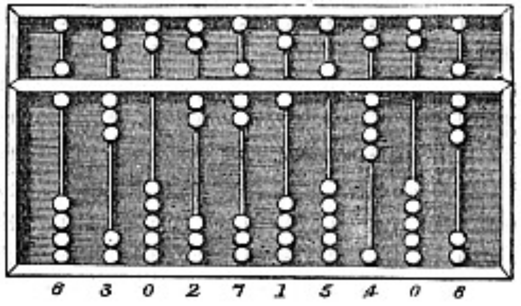
- 3,000 years ago: Mathematics, logic, and numerical computation
  - Important contributions made by the Greeks, Egyptians, Babylonians, Indians, Chinese, and Persians
  - Cuneiform
  - Stone “abacus”
- http://www.thocp.net/slideshow/0469.htm

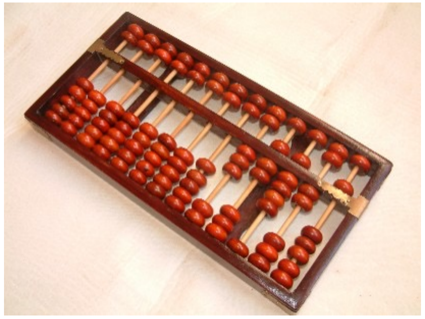
Slide from “Introduction to Computing”

# ABACUS

Early calculating devices

ABACUS – 2700 BC  (Mesopotamia)



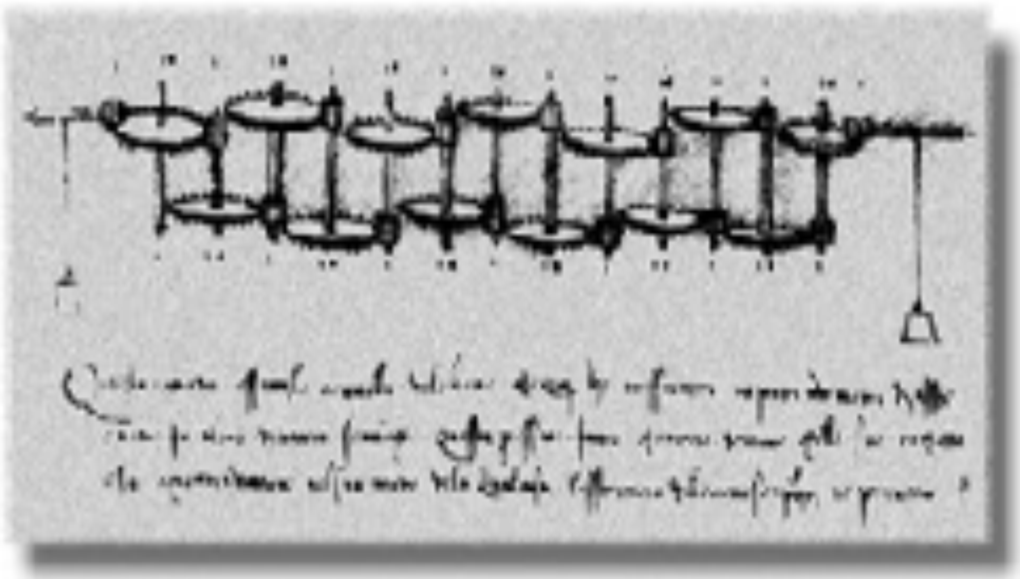
http://www.computersciencelab.com/ComputerHistory/History.htm

Slide from “Introduction to Computing”

METU Computer Engineering

## DaVinci

- 1452-1519 Leonardo DaVinci sketched gear-driven calculating machines but none were ever built.



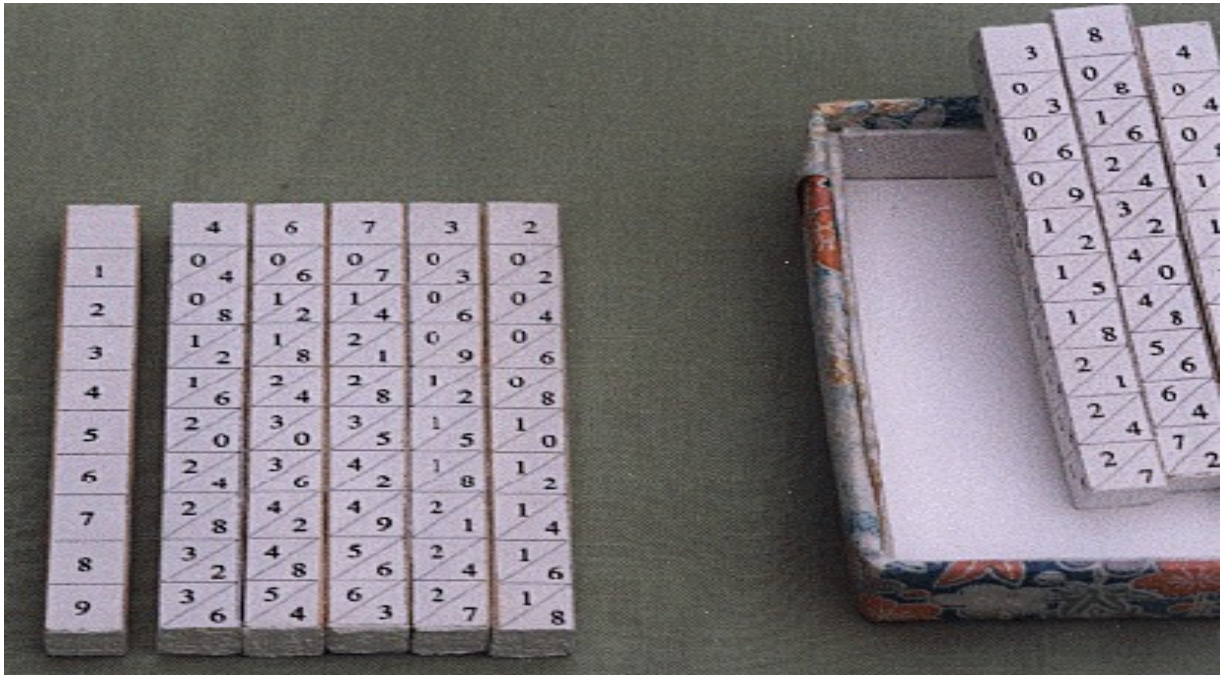
http://www.computersciencelab.com/ComputerHistory/History.htm

Slide from "Introduction to Computing"

# Napier's Bones

- 1614: Logarithms
  - Invented by John Napier to simplify difficult mathematical computations

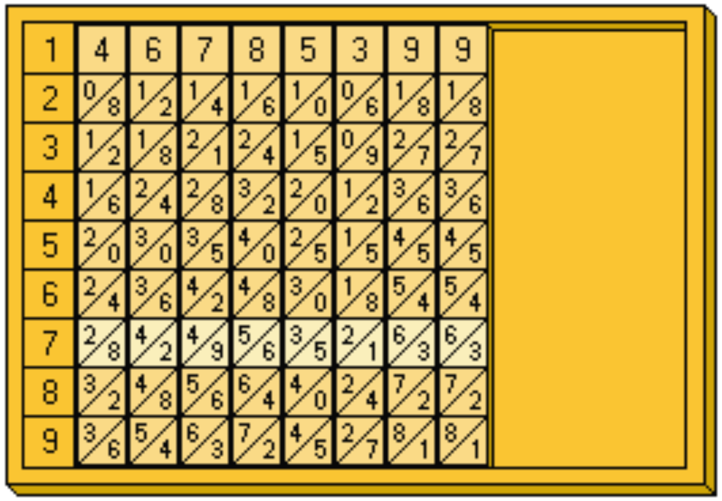
Napier's Bones:



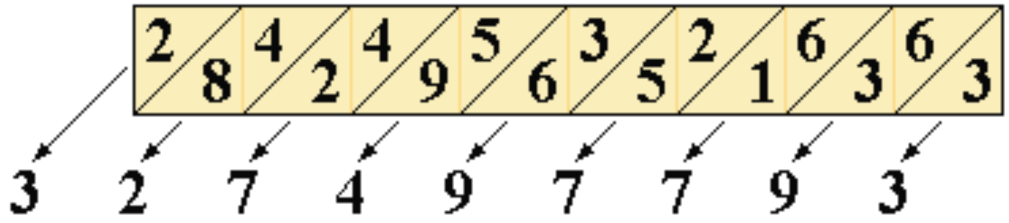
http://www.computersciencelab.com/ComputerHistory/History.htm

Slide from "Introduction to Computing"

- If you want to multiply 7 by 46785499:



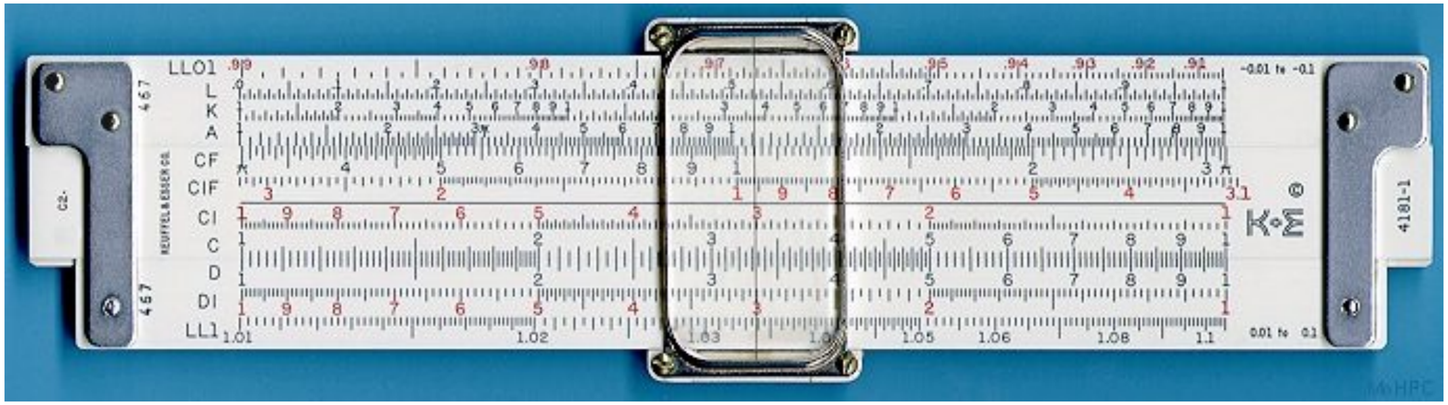
| 1 | 4 | 6 | 7 | 8 | 5 | 3 | 9 | 9 |
|---|---|---|---|---|---|---|---|---|
| 2 | 0/8 | 1/2 | 1/4 | 1/6 | 1/0 | 0/6 | 1/8 | 1/8 |
| 3 | 1/2 | 1/8 | 2/1 | 2/4 | 1/5 | 0/9 | 2/7 | 2/7 |
| 4 | 1/6 | 2/4 | 2/8 | 3/2 | 2/0 | 1/2 | 3/6 | 3/6 |
| 5 | 2/0 | 3/0 | 3/5 | 4/0 | 2/5 | 1/5 | 4/5 | 4/5 |
| 6 | 2/4 | 3/6 | 4/2 | 4/8 | 3/0 | 1/8 | 5/4 | 5/4 |
| 7 | 2/8 | 4/2 | 4/9 | 5/6 | 3/5 | 2/1 | 6/3 | 6/3 |
| 8 | 3/2 | 4/8 | 5/6 | 6/4 | 4/0 | 2/4 | 7/2 | 7/2 |
| 9 | 3/6 | 5/4 | 6/3 | 7/2 | 4/5 | 2/7 | 8/1 | 8/1 |



METU Computer Engineering

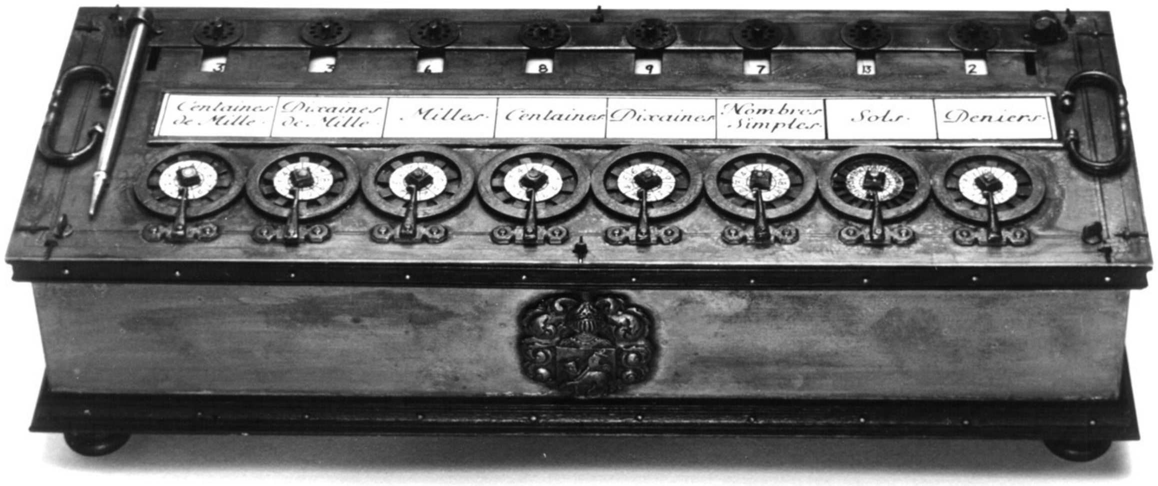
# Slide Rule (slipstick) "a mechanical analog computer"

Around 1622: First slide rule created



http://www.computersciencelab.com/ComputerHistory/History.htm

Slide from "Introduction to Computing"

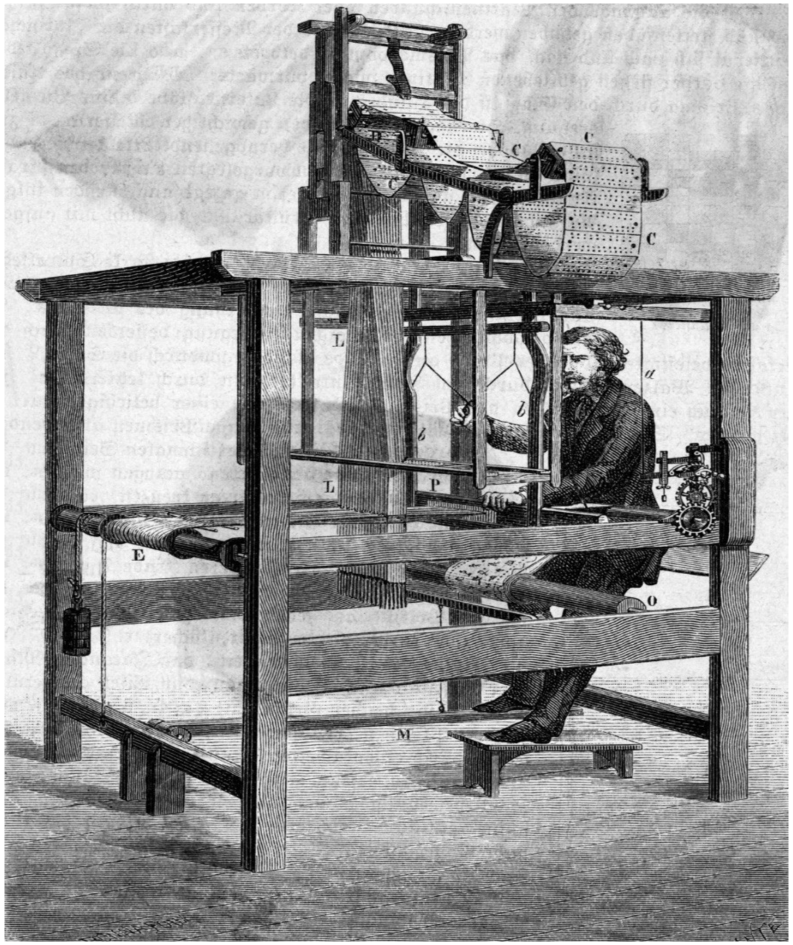


## The Pascaline: One of the Earliest Mechanical Calculators

Slide from "Introduction to Computing"

METU Computer Engineering

## The Early Period: Up to 1940

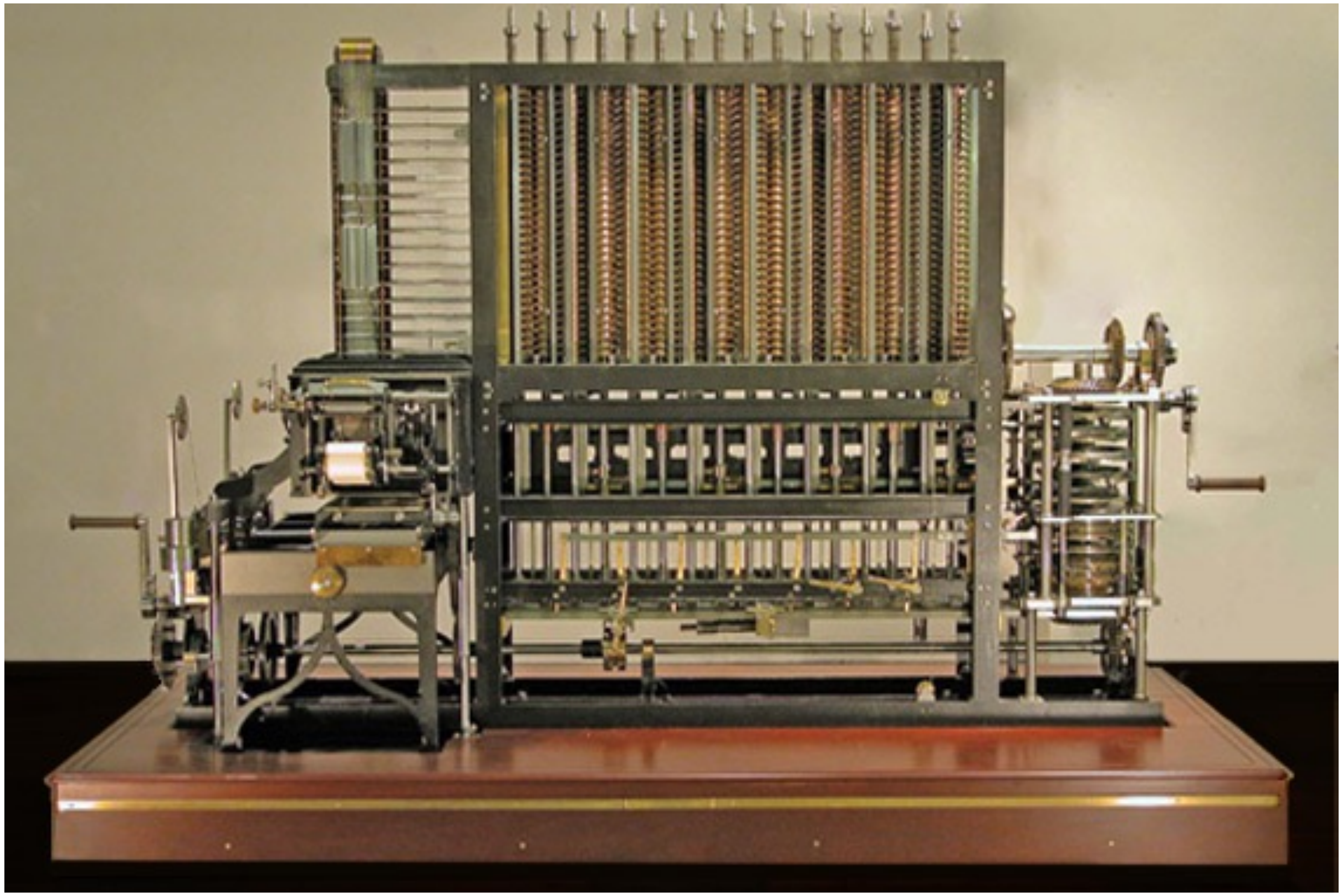


Jacquard's Loom

Also see http://www.computersciencelab.com/ComputerHistory/HistoryPt2.htm

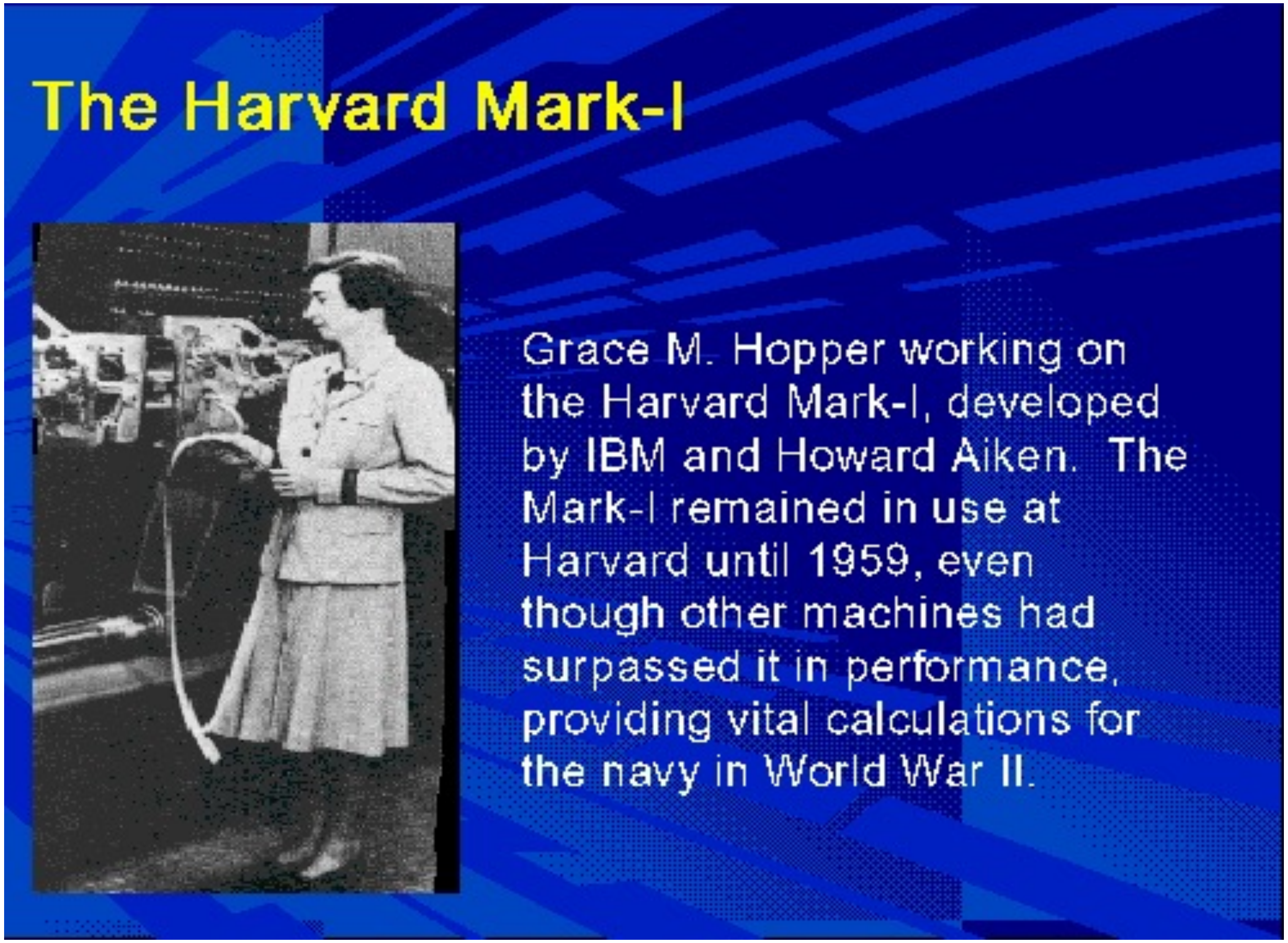
Slide from "Introduction to Computing"

## Difference engine



http://www.youtube.com/watch?v=0anIyVGeWOI

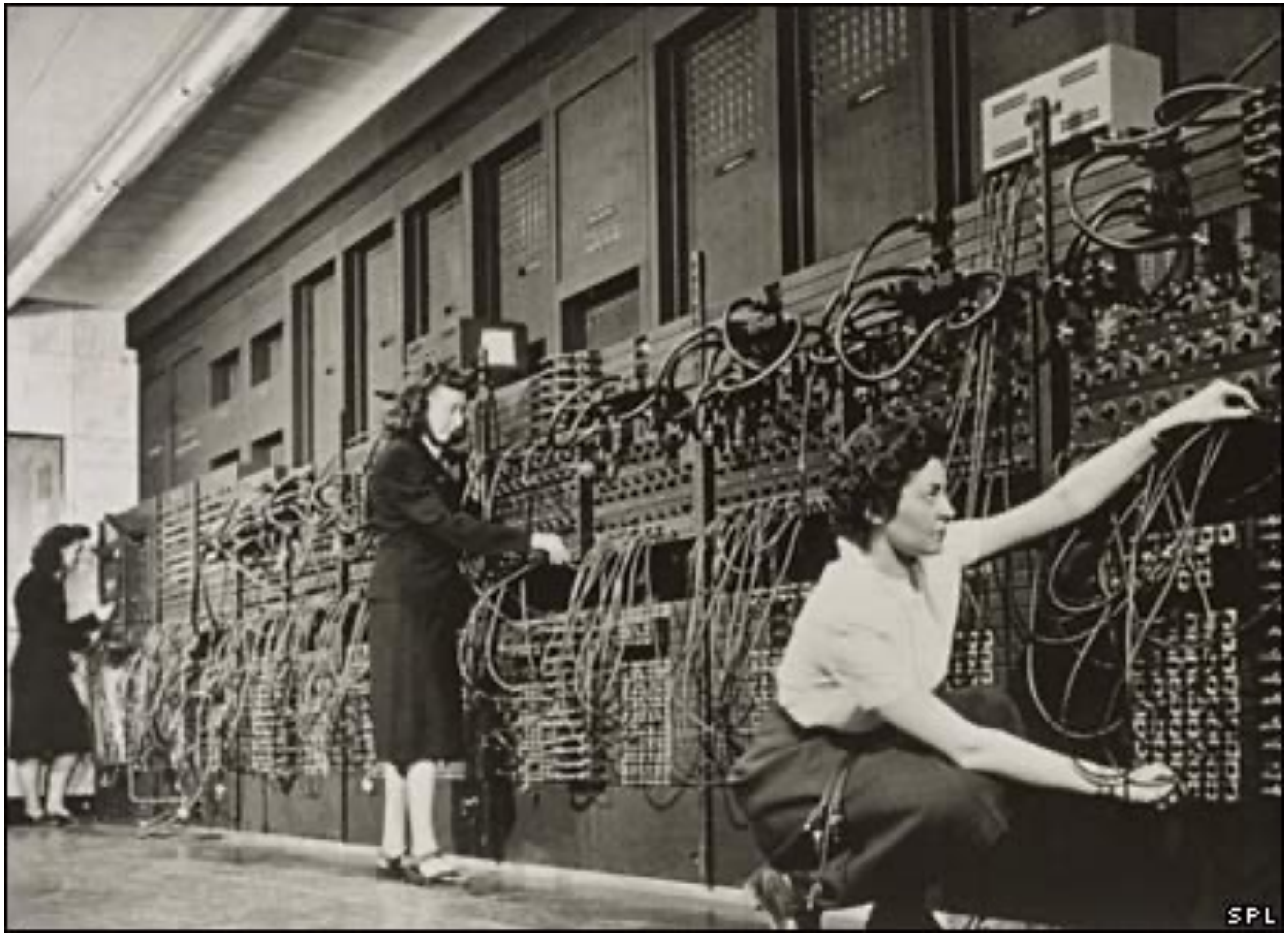
Slide from "Introduction to Computing"



## The Harvard Mark-I

Grace M. Hopper working on the Harvard Mark-I, developed by IBM and Howard Aiken. The Mark-I remained in use at Harvard until 1959, even though other machines had surpassed it in performance, providing vital calculations for the navy in World War II.

Slide from "Introduction to Computing"

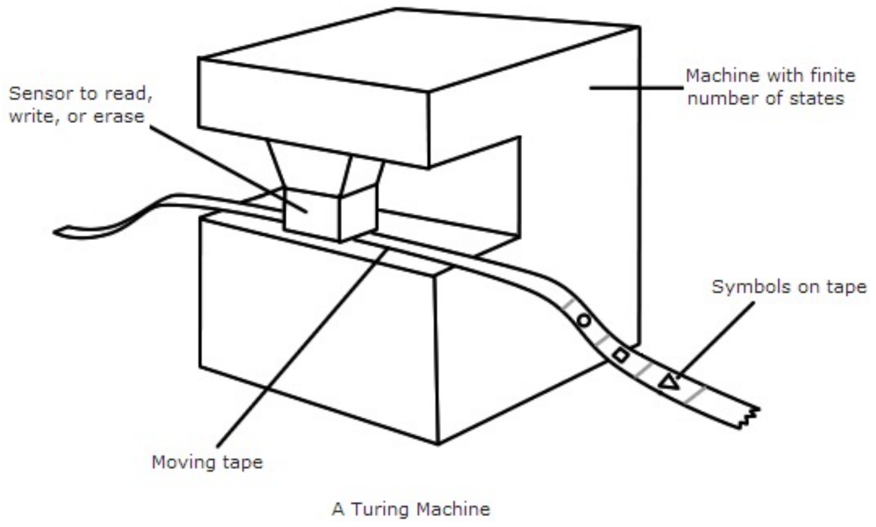


## Programming the ENIAC

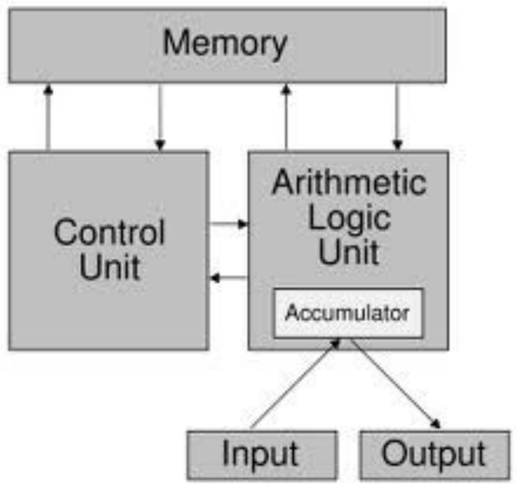
Slide from "Introduction to Computing"

## History of Computation

- Read the reading material on this subject!
- And watch a video whose link we will post on odtuclass.


Sensor to read, write, or erase
Machine with finite number of states
Symbols on tape
Moving tape
A Turing Machine

Turing Machine

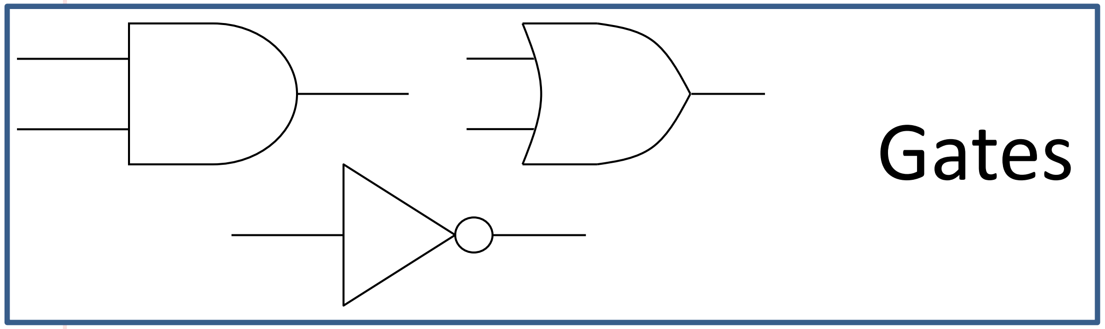


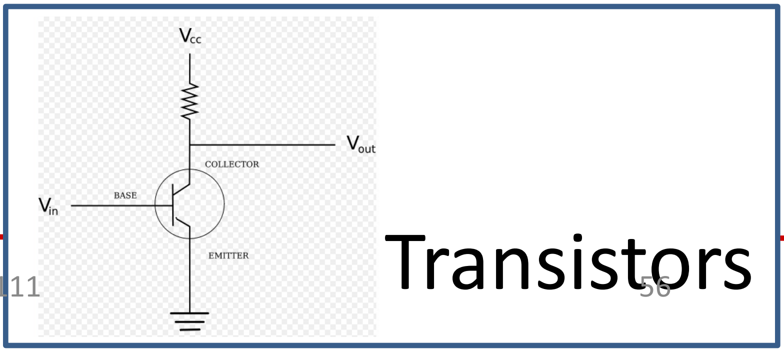
Von Neumann Architecture

# DIGITAL COMPUTATION

# A computer

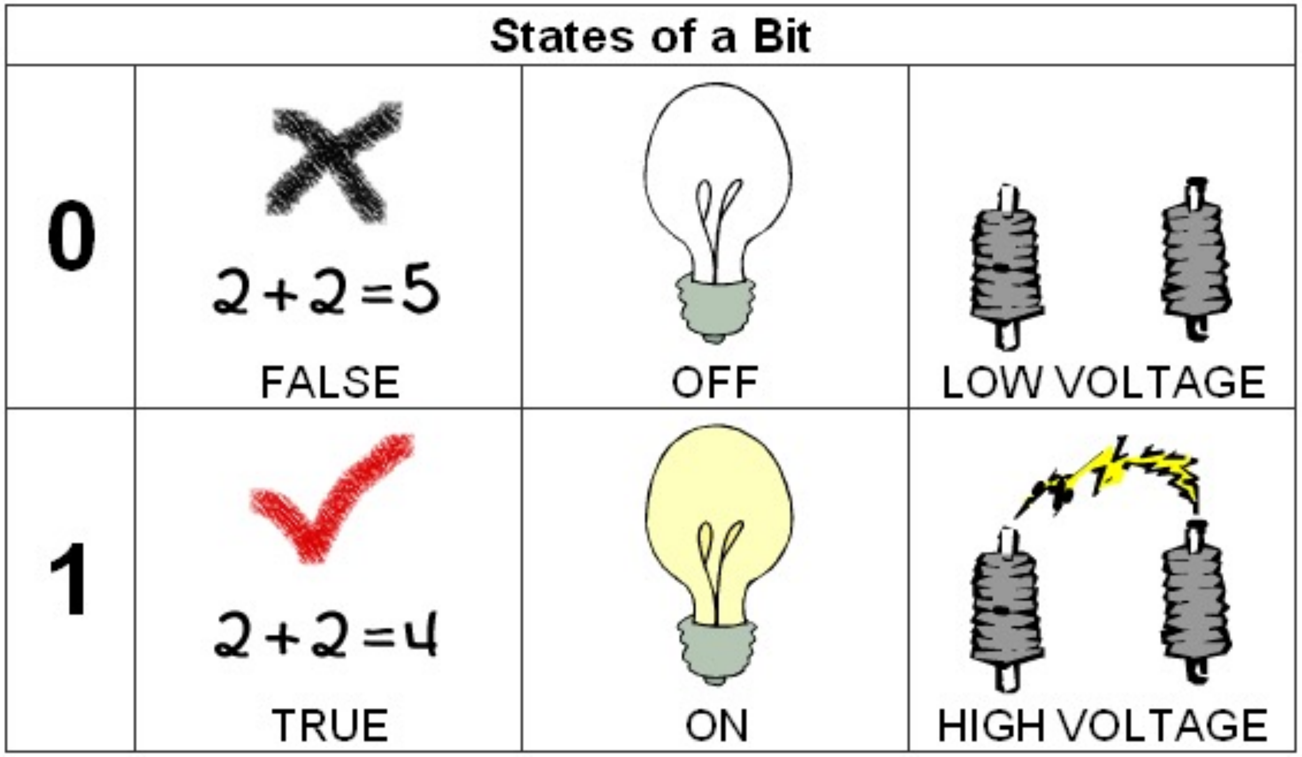
Devices


Gates



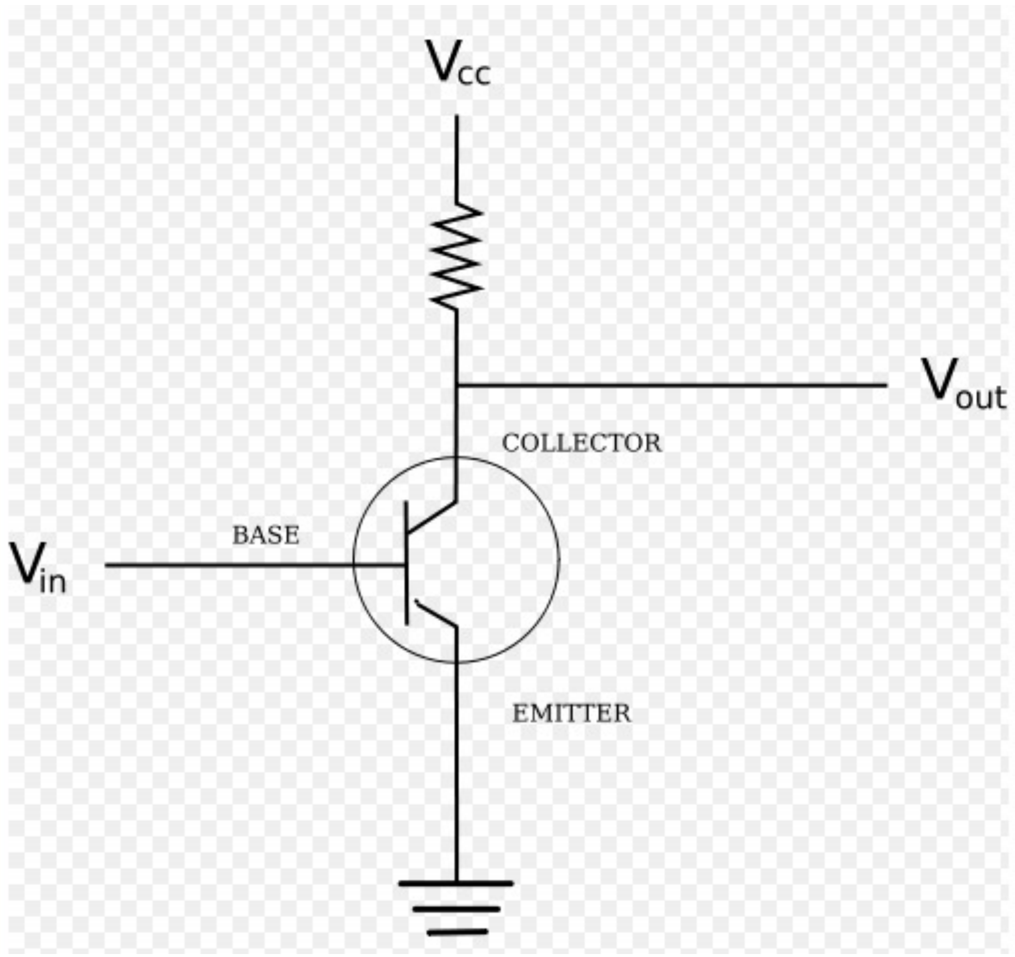
Transistors

# Everything in a PC is Binary … well, almost …



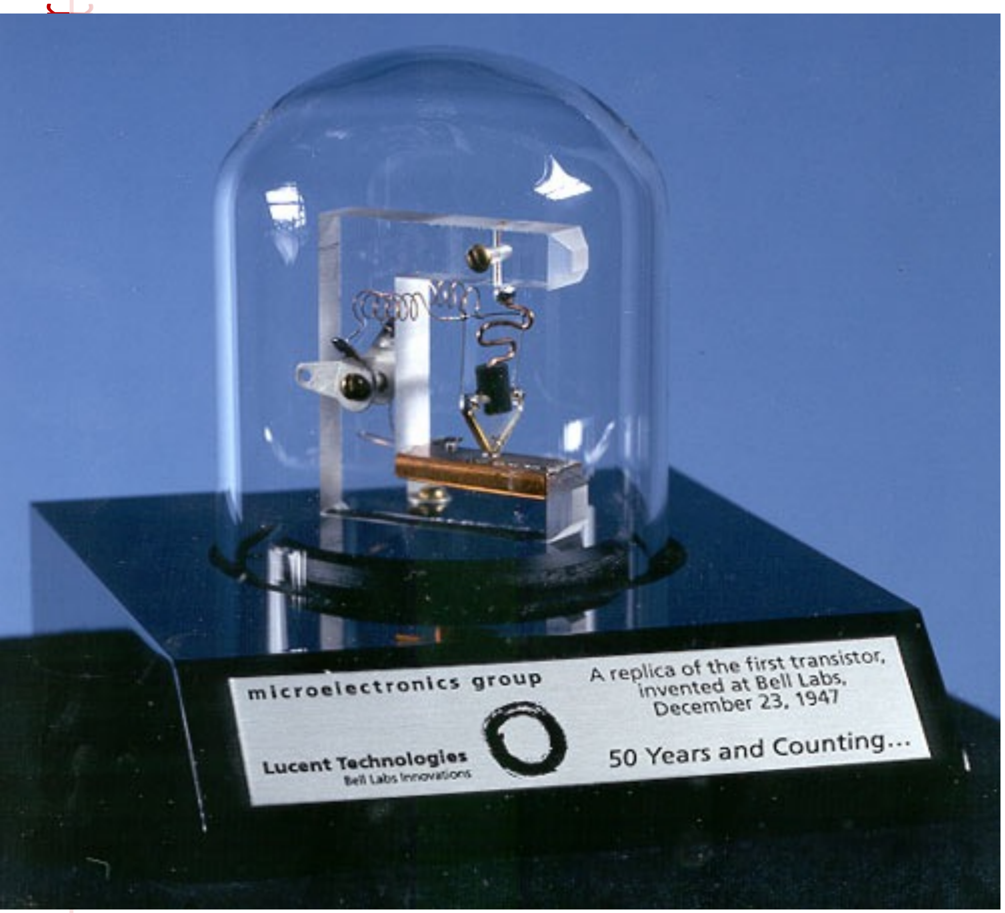
| States of a Bit | | | |
|---|---|---|---|
| 0 | 2+2=5 FALSE | OFF | LOW VOLTAGE |
| 1 | 2+2=4 TRUE | ON | HIGH VOLTAGE |

## A transistor



This circuit functions as a switch. In other words, based on the *control* voltage, the circuit either passes Vin to output or not.

# Examples of transistors



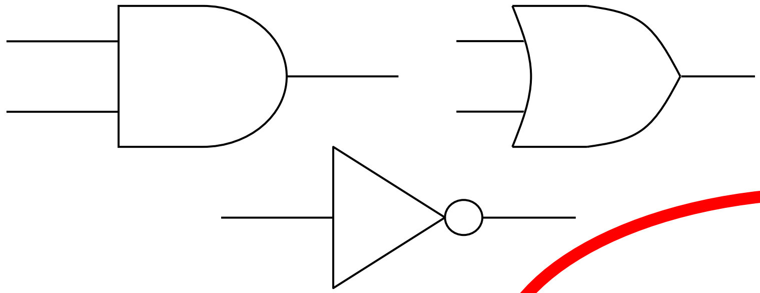
Replica of the first transistor

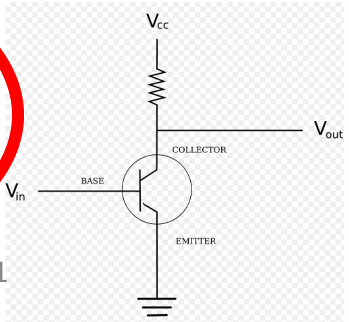
A set of transistors, depicting the fast change in technology.

# A computer

Devices

Gates



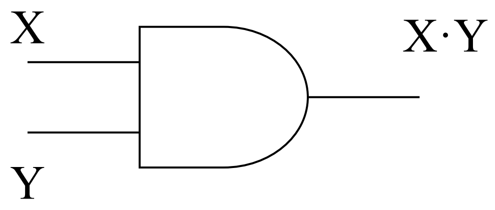


Transistors

METU Computer Engineering

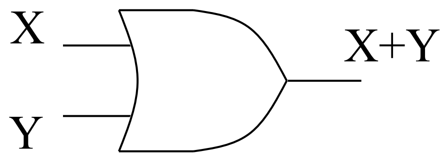
# AND gate

| X | Y | X·Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



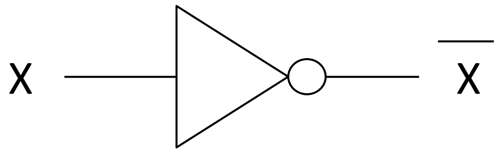
## OR Gate

| X | Y | X+Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



# NOT Gate

| X | $\overline{X}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |



# XOR Gate

| X | Y | X⊕Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |