Introduction to Computer and Programming

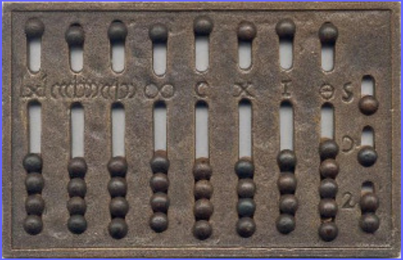
Chapter 1: Computer and Programming

Manuel

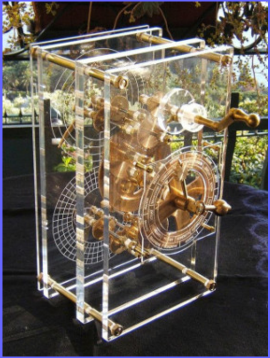
Fall 2018

Outline

Ancient Era

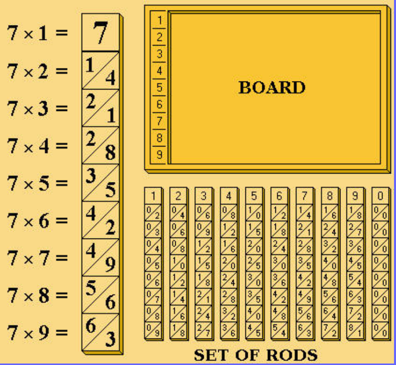


Abacus (-2700)



Antikythera mechanism (-100)

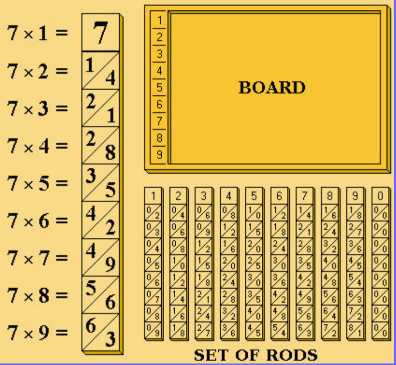
Calculating Tools



Napier's bones (1617)

Sliderule (1620)

Calculating Tools



Napier's bones (1617)

Sliderule (1620)

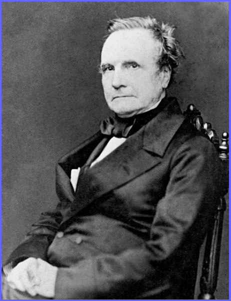
Note: first pocket calculator around 1970 in Japan.

Mechanical Calculators

Pascaline (1642)

Arithmomètre (1820)

The 19th Century



Charles Babbage (1791–1871)

- Difference Engine (Built in the 1990es)
- Analytical Engine (Never built)

Ada Byron (1815–1852)

- Extensive notes on Babbage work
- Algorithm to calculate a sequence of Bernoulli numbers using the Analytical Engine

Birth of Modern Computing

First part of the 20th century:

- **1936:** First freely programmable computer
- **1946:** First electronic general-purpose computer
- **1948:** Invention of the transistor
- **1951:** First commercial computer
- **1958:** Integrated circuit

UNIVAC I (1951)

Toward Modern Computing

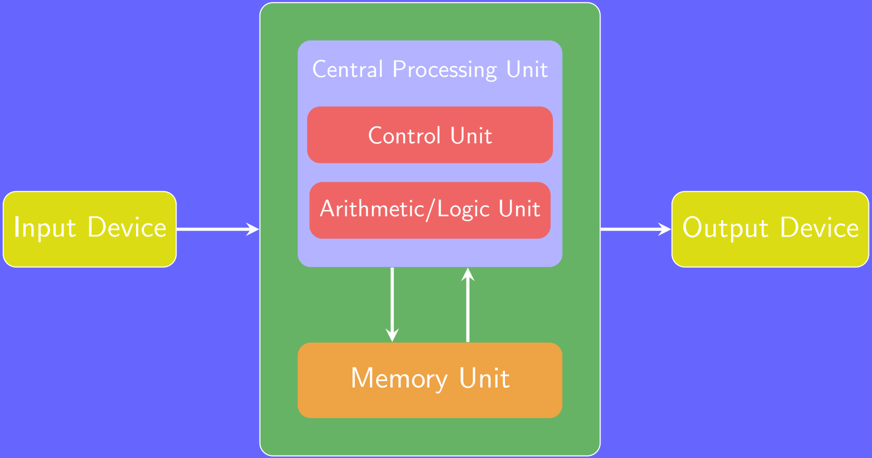
Apple I (1976)

Second part of the 20th century:

- **1962:** First computer game
- **1969:** ARPAnet
- **1971:** First microprocessor
- **1975:** First consumer computers
- **1981:** First PC, MS-DOS
- **1983:** First home computer with a GUI
- **1985:** Microsoft Windows
- **1991:** Linux

Outline

Von Neumann architecture



What does a computer understand?

Numbers in various bases:

- Humans use *decimal* $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, e.g. $(253)_{10}$
- Computers work internally using *binary* (0,1), e.g $(11111101)_2$
- Human-friendly way to represent binary: *hexadecimal* (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), e.g. $(\text{FD})_{16}$

Number base conversion

Base conversion:

- From base b into decimal: evaluate the polynomial
  $(11111101)_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 253$
  $(\mathsf{FD})_{16} = \mathsf{F} \cdot 16^1 + \mathsf{D} \cdot 16^0 = 15 \cdot 16^1 + 13 \cdot 16^0 = 253$
- From decimal into base b: repeatedly divide n by b until the quotient is 0. Consider the remainders from right to left
  rem(253,2)=1, rem(126,2)=0, rem(63,2)=1, rem(31,2)=1, rem(15,2)=1, rem(7,2)=1, rem(3,2)=1, rem(1,2)=1
  rem(253,16)=13=D, rem(15,16)=15=F
- From base b into base b^a: group numbers into chunks of a elements
  $(11111101)_2 = 1111\ 1101 = (\mathsf{FD})_{16}$

Quick examples

Exercise.

- Convert into hexadecimal: 1675, 321, $(100011)_2$, $10111011)_2$
- Convert into binary: 654, 2049, ACE, 5F3EC6
- Convert into decimal: $(111110)_2$, $(10101)_2$, $(12345)_{16}$, 12C3C

Quick examples

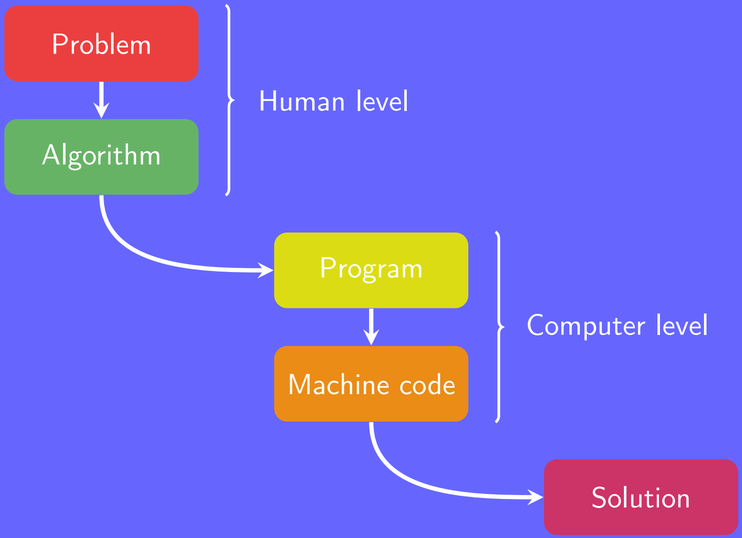
Exercise.

- Convert into hexadecimal: 1675, 321, $(100011)_2$, $10111011)_2$
- Convert into binary: 654, 2049, ACE, 5F3EC6
- Convert into decimal: $(111110)_2$, $(10101)_2$, $(12345)_{16}$, 12C3C

Solution.

$1675 = 68\text{B},\ 321 = (141)_{16},\ (100011)_2 = (23)_{16},$
$654 = (1010001110)_2,\ 2049 = (100000000001)_2,$
$\text{ACE} = 101011001110,\ 5\text{F}3\text{EC}6 = (10111110011111011000110)_2$
$(111110)_2 = 62,\ (10101)_2 = 21,\ (12345)_{16} = 74565,$
$12\text{C}3\text{C} = 76860$

How to use a computer?



Algorithm

Algorithm: recipe telling the computer how to solve a problem.

Algorithm

Algorithm: recipe telling the computer how to solve a problem.

Example.
I am the "computer", detail an algorithm such that I can prepare a jam sandwich.

Actions: cut, listen, spread, sleep, read, take, eat, dip, assemble
Things: knife, guitar, bread, honey, jamjar, sword, slice

Algorithm

Algorithm: recipe telling the computer how to solve a problem.

Example.
I am the "computer", detail an algorithm such that I can prepare a jam sandwich.

Actions: cut, listen, spread, sleep, read, take, eat, dip, assemble
Things: knife, guitar, bread, honey, jamjar, sword, slice

Algorithm. (*Sandwich making*)

Input : 1 bread, 1 jamjar, 1 knife
Output: 1 jam sandwich

take the knife and cut 2 slices of bread;
dip the knife into the jamjar;
spread the jam on the bread, using the knife;
assemble the 2 slices together, jam on the inside;

Program

Algorithm vs **Machine code**

$$\begin{tikzcd}
\text{Algorithm} \arrow[r] & \text{Programming Language} \arrow[d] \\
\text{Machine Code} & \text{Interpreter / Compiler} \arrow[l]
\end{tikzcd}$$

Example

Problem: given a square and the length of one side, what is its area?

Algorithm.

Input : side (the length of one side of a square)
Output : the area of the square

return side $*$ side

area.c

```c
#include<stdio.h>
int main() {
  int side;
  printf("Side: ");
  scanf("%d",&side);
  printf("Area: %d",\
    side*side);
}
```

area.cpp

```cpp
#include <iostream>
using namespace std;
int main() {
int side;
  cout << "Side: ";
  cin >> side;
  cout << "Area: "\
    << side*side;
  return 0;
}
```

area.m

```matlab
a=input("Side: ");
printf ("Area: %d",...
  a*a)
```

Running the program

To see the result of a program:

- C or C++
  1. Write the source code
  2. Compile the program
  3. Run the program
- MATLAB
  1. Type the code
  2. Press `Return`

Outline

Mathematical software

Common math software:

- Axiom
- GAP
- gp
- Magma
- Maple
- Mathematica
- MATLAB
- Maxima
- Octave
- R
- Scilab

MATLAB

MATLAB=MATrix LABoratory

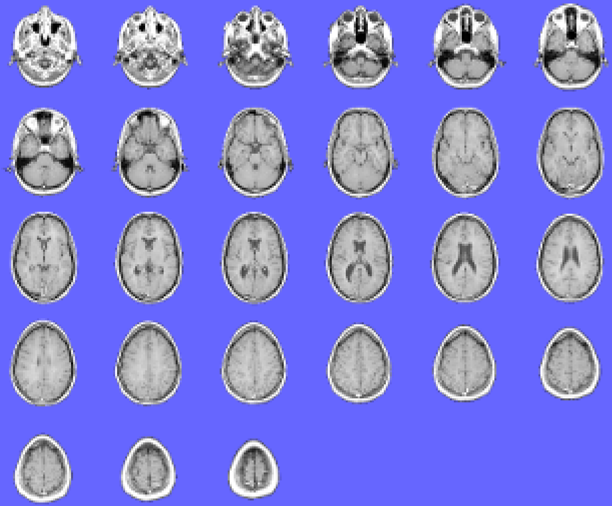
- **Matrix manipulations**
- **Implement algorithms**
- **Plotting functions/data**
- Create user interfaces
- Interfaced with other programming languages

Why MATLAB?

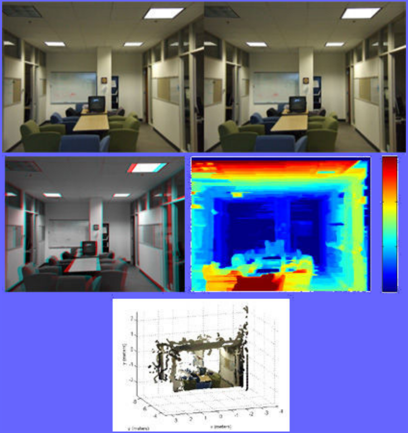
Engineers like MATLAB:

- Easy to use
- Versatile
- Built-in programming languages
- Many toolboxes
- Widely used in academia and industry

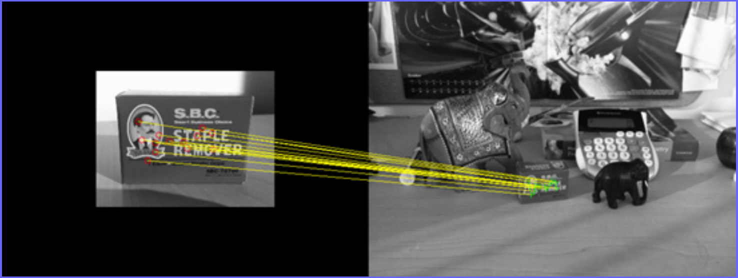
MRI slices



Stereo Vision



Object detection



Mathematics and Physics

$$\begin{tikzcd}
\text{Physical problem} \arrow[r] & \text{Mathematical problem} \arrow[d] \\
\text{Physical solution} \arrow[u] & \text{Mathematical solution} \arrow[l]
\end{tikzcd}$$

What to do?

Before jumping on the computer and start coding:

- Clearly state/translate the problem
- What is known $\longrightarrow$ INPUT
- What is to be found $\longrightarrow$ OUTPUT
- Find a systematic way to solve the problem $\longrightarrow$ Algorithm
- Check the solution
- Start implementing

Example

Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

Studying the problem

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

- Easy part
    - Problem: finding the density of the sun
    - Initial input: distance r, circumference c
    - Output: density d

Studying the problem

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

- Easy part
  - Problem: finding the density of the sun
  - Initial input: distance r, circumference c
  - Output: density d
- Potentially more complicated part
  1. Density
  2. Sun $\sim$ sphere, $radius = \frac{circumference}{2\pi} \Rightarrow$ volume V
  3. Mass of the sun:

Studying the problem

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

- Easy part
  - Problem: finding the density of the sun
  - Initial input: distance r, circumference c
  - Output: density d
- Potentially more complicated part
  1. Density
  2. Sun $\sim$ sphere, $radius = \frac{circumference}{2\pi} \Rightarrow$ volume V
  3. Mass of the sun: Kepler's 3rd law: $\frac{T^2}{r^3} = \frac{4\pi^2}{GM}$
  4. $M = \frac{4\pi^2 r^3}{GT^2}$

The Algorithm

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

Algorithm.

Input : $r = 1.496 \cdot 10^8,\ c = 4.379 \cdot 10^6,\ G = 6.674 \cdot 10^{-11},$
$T = 365\text{D}$

Output : Density of the Sun

$V \leftarrow \frac{4}{3}\pi(\frac{c}{2\pi})^3$;

$M \leftarrow \frac{4\pi^2 r^3}{GT^2}$;

return $\frac{M}{V}$;

The Algorithm

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

Algorithm.

Input : $r = 1.496 \cdot 10^8,\ c = 4.379 \cdot 10^6,\ G = 6.674 \cdot 10^{-11}$,
$T = 365\mathrm{D}$

Output : Density of the Sun

$V \leftarrow \frac{4}{3}\pi(\frac{c}{2\pi})^3$;
$M \leftarrow \frac{4\pi^2 r^3}{GT^2}$;
return $\frac{M}{V}$;

Run the algorithm: 338110866080

WRONG!

WRONG!

UNITS...

The Algorithm

Problem: Given that the sun is located $1.496 \cdot 10^8$ km away from the Earth and has a circumference of $4.379 \cdot 10^6$ km, calculate its density.

Algorithm.

Input : $r = 1.496 \cdot 10^{11}$ m, $c = 4.379 \cdot 10^9$ m,
$G = 6.674 \cdot 10^{-11}$ m^3/kg/s^2, $T = 365 * 24 * 3600$ s

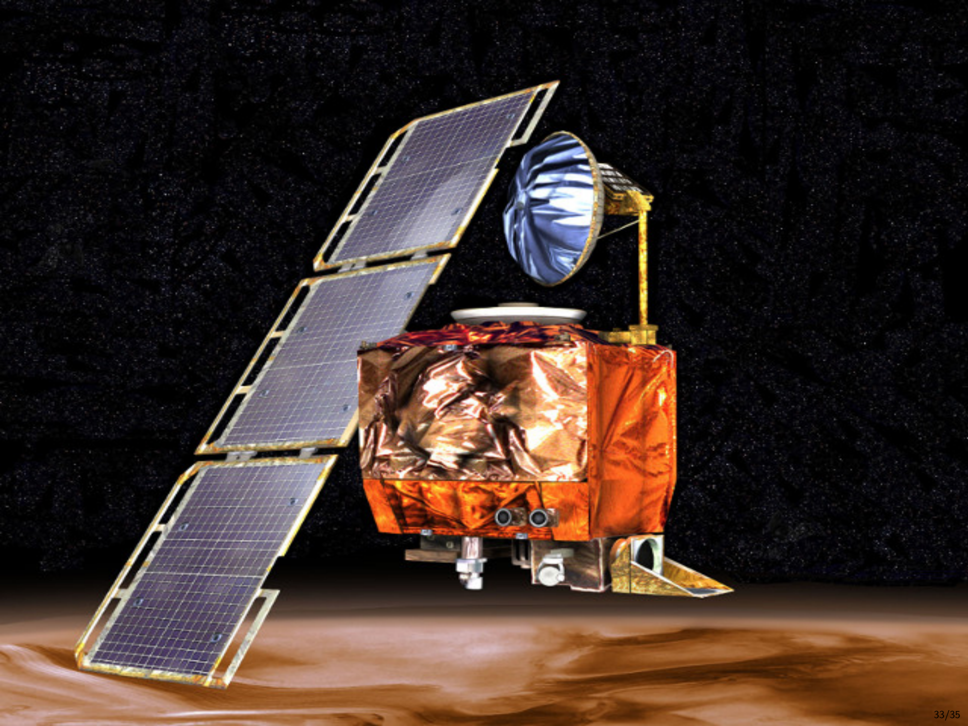
Output : Density of the Sun

$V \leftarrow \frac{4}{3}\pi(\frac{c}{2\pi})^3$;

$M \leftarrow \frac{4\pi^2 r^3}{GT^2}$;

return $\frac{M}{V}$;

Run the algorithm: 1404 kg/m^3



Key points

- What is a programming language?
- What are the two main types of programming language?
- What is an algorithm?
- How to tackle a problem?

Thank you!