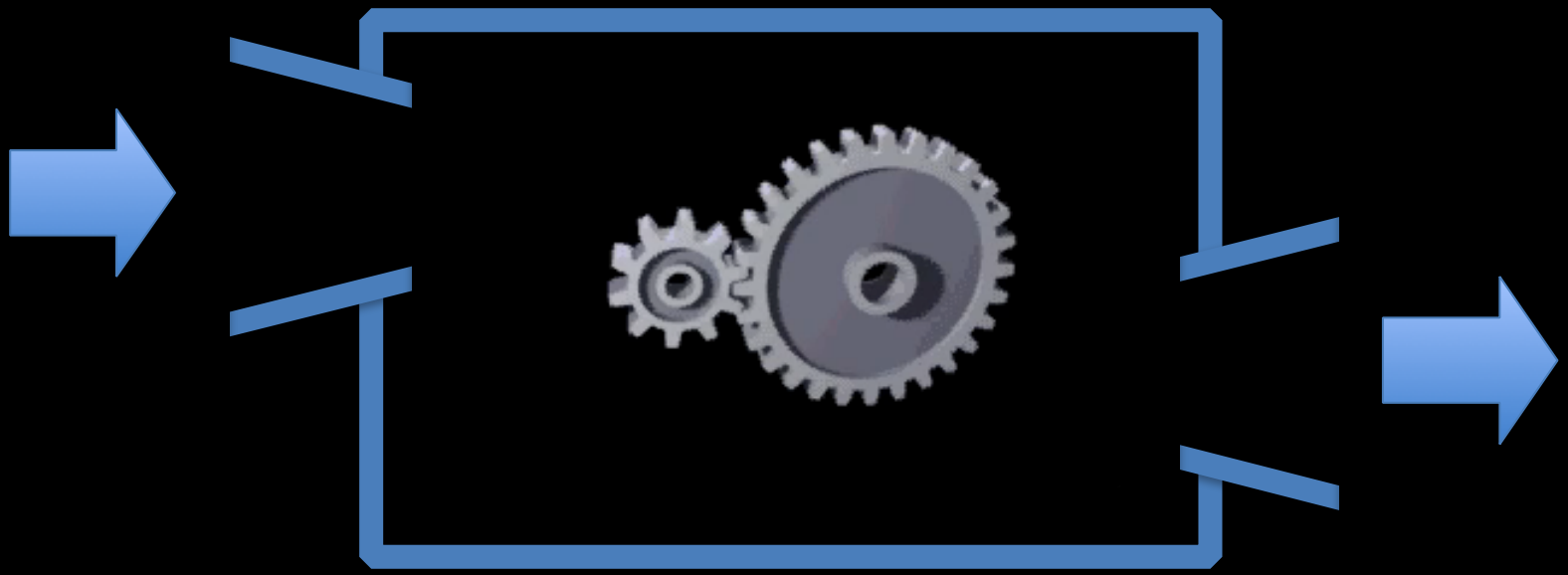
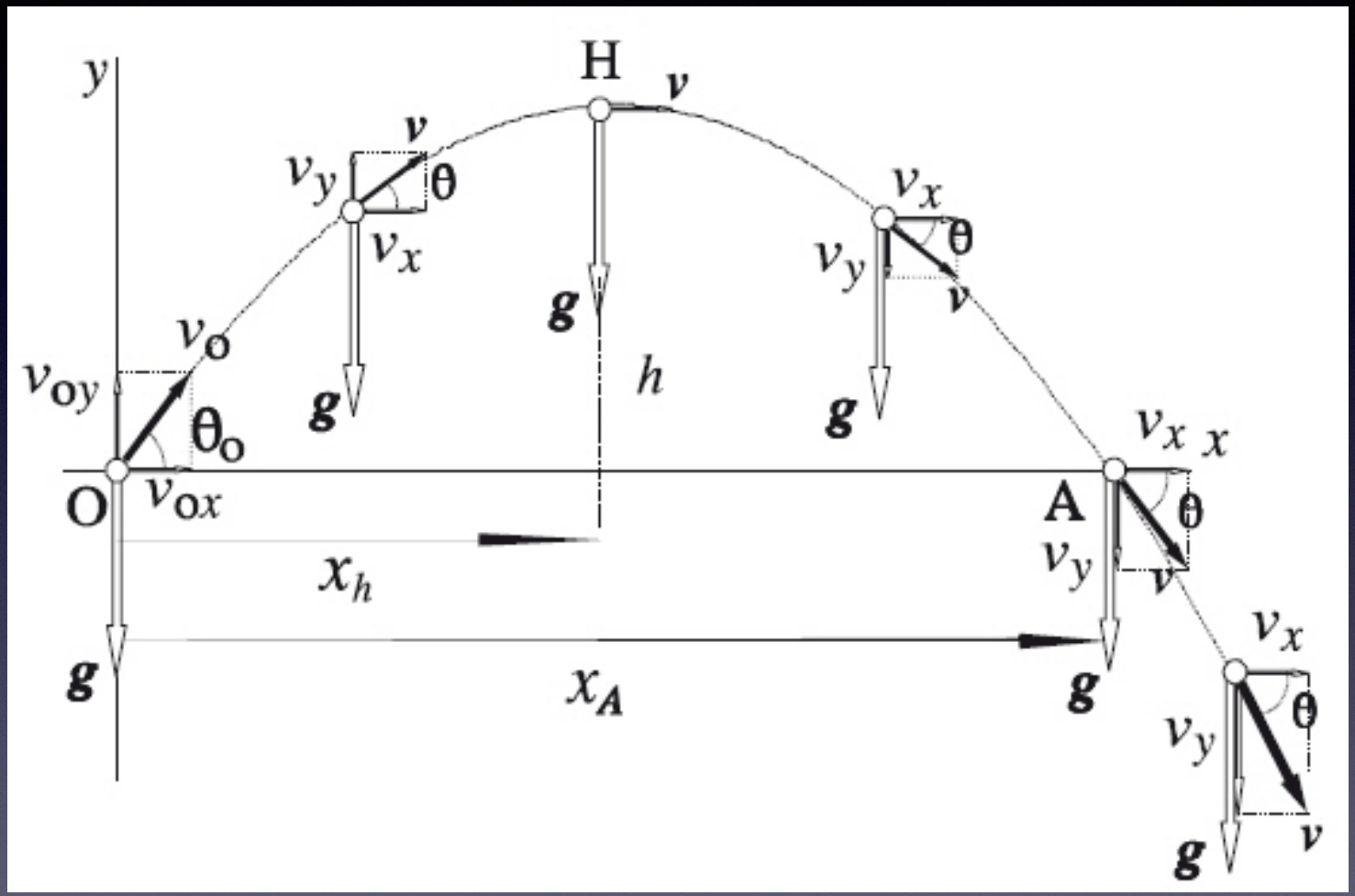
Algoritmos
Introdução a Computação

Prof. Hitoshi Nagano, Ph.D.

Aula 0




Cerantior
8
TRK Kyros
7
Dorigem1
9
acensa
MataVelinhas
7
playerwhitemer
8
Ajnagano
8
BrEpidemic
8


y
H
v
v_y
v
θ
v_x
v_x
θ
v_y
v
g
h
g
g
v_o
v_{oy}
θ_o
O
v_{ox}
x_h
A
v_x x
θ
v_y
v
g
x_A
g
v_x
θ
v_y
v
g

Ciencia da Computação

Computação

- Que problemas
- Como resolver
- Quais técnicas

áreas

- Sistemas
- Inteligencia Artificial
- Segurança
- Redes
- Linguagens
- Teoria
- Aplicações Científicas

Como…

- Abstração
- Gestão da complexidade

Técnicas

- Paradigmas de programação
- Lingugem C, principalmente

O que esperar…

- estudo e prática
- provas individuais e trabalhos em grupo
- comunidade, colaboração, discussão
- dedicação
- … recompensas
- … Satisfaction

Ciencia da Computação

O início

Sistema

- Binario: `0, 1`
- Decimal: `0, 1, 2, 3, 4, 5, 6, 7, 8, 9`
- Hexadecimal: `0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f`

6811

- Binario: `0b0001101010011011`
- Decimal: `6811`
- Octal: `0o15233`
- Hexadecimal: `0x1a9b`

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31
100000	32
&c.	&c.

bres entiers au-deſſous du double du plus haut degré. Car icy, c'eſt comme ſi on diſoit, par exemple, que 111 ou 7 eſt la ſomme de quatre, de deux & d'un.

1001	4
10	2
1	1
111	7

Et que 1101 ou 13 eſt la ſomme de huit, quatre & un. Cette proprieté ſert aux Eſſayeurs pour peſer toutes ſortes de maſſes avec peu de poids, & pourroit ſervir dans les monnoyes pour donner pluſieurs valeurs avec peu de pieces.

1000	8
100	
1	1
1101	13

Cette expreſſion des Nombres étant établie, ſert à faire tres-facilement toutes ſortes d'operations.

Pour *l'Addition* par exemple. ☽

110	6	101	5	1110	14
111	7	1011	11	10001	17
1101	13	10000	16	11111	31

Pour la *Souſtraction.*

1101	13	10000	16	11111	31
111	7	1011	11	10001	17
110	6	101	5	1110	14

Pour la *Multiplication.* ⊙

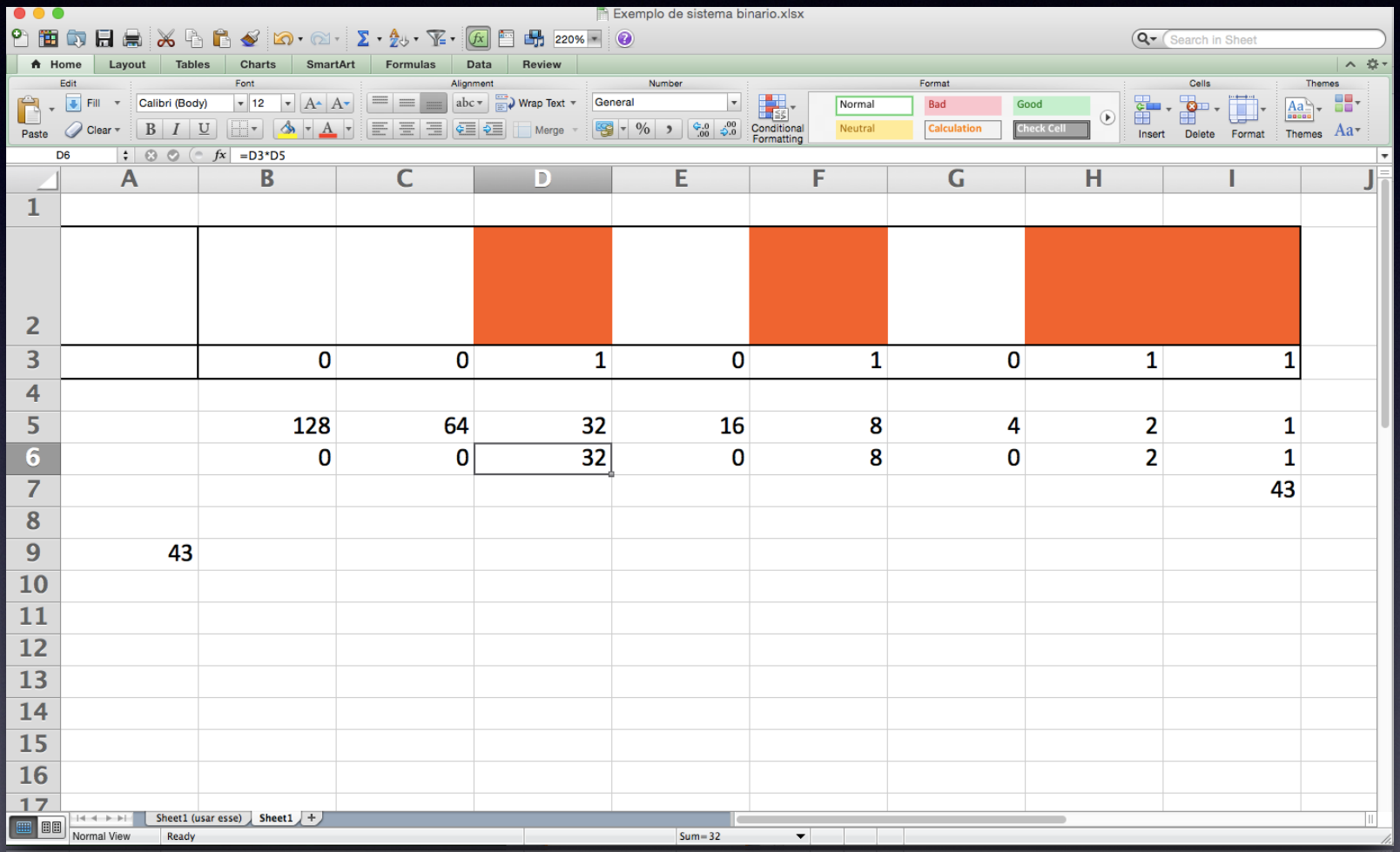
11	3	101	5	101	5
11	3	11	3	101	5
11		101		101	
11		101		1010	
1001	9	1111	15	11001	25

Pour la *Diviſion.*

15	x x 1 1	)	101 | 5
3	x x x 1		
	x 1		

Et toutes ces operations ſont ſi aiſées, qu'on n'a jamais beſoin de rien eſſayer ni deviner, comme il faut faire dans la diviſion ordinaire. On n'a point beſoin non-plus de rien apprendre par cœur icy, comme il faut faire dans le calcul ordinaire, où il faut ſçavoir, par exemple, que 6 & 7 pris enſemble font 13; & que 5 multiplié par 3 donne 15, ſuivant la Table *d'une fois un eſt un*, qu'on appelle Pythagorique. Mais icy tout cela ſe trouve & ſe prouve de ſource, comme l'on voit dans les exemples précedens ſous les ſignes ☽ & ⊙.

"Explication de l'Arithmétique Binaire"
Leibniz



	A	B	C	D	E	F	G	H	I	J
1										
2										
3		0	0	1	0	1	0	1	1	
4										
5		128	64	32	16	8	4	2	1	
6		0	0	32	0	8	0	2	1	
7									43	
8										
9	43									
10										
11										
12										
13										
14										
15										
16										
17										

xxd

Mostra o conteúdo dos arquivos (no LINUX)

arquivo bmp

```
xxd -c 24 -g 3 -s 54 carafeliz.bmp
```

arquivo bmp

```
0000036: ffffff ffffff 0000ff 0000ff 0000ff 0000ff ffffff ffffff ..........
000004e: ffffff 0000ff ffffff ffffff ffffff ffffff 0000ff ffffff ..........
0000066: 0000ff ffffff 0000ff ffffff ffffff 0000ff ffffff 0000ff ..........
000007e: 0000ff ffffff ffffff ffffff ffffff ffffff ffffff 0000ff ..........
0000096: 0000ff ffffff 0000ff ffffff ffffff 0000ff ffffff 0000ff ..........
00000ae: 0000ff ffffff ffffff 0000ff 0000ff ffffff ffffff 0000ff ..........
00000c6: ffffff 0000ff ffffff ffffff ffffff ffffff 0000ff ffffff ..........
00000de: ffffff ffffff 0000ff 0000ff 0000ff 0000ff ffffff ffffff ..........
```

arquivo bmp

A) Cabeçalho de arquivo – informações do arquivo - Tamanho : **14 bytes**

B) Cabeçalho de mapa de bits – informações da imagem - Tamanho : **40 bytes**

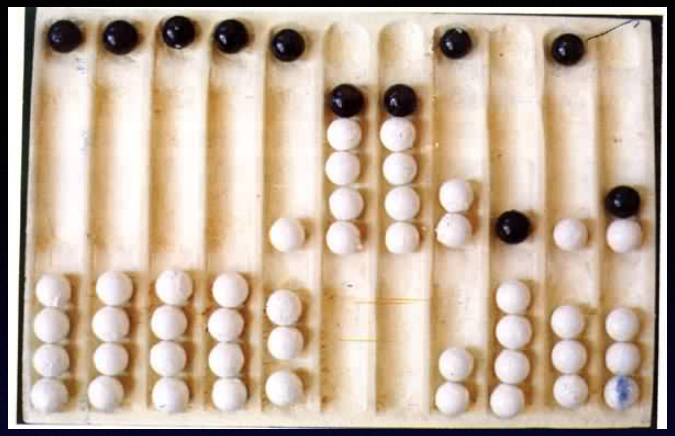
Fonte: http://www2.ic.uff.br/~aconci/curso/bmp.pdf

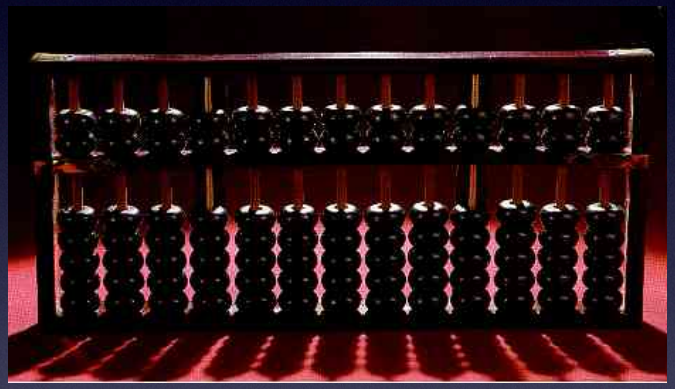
C ?

C

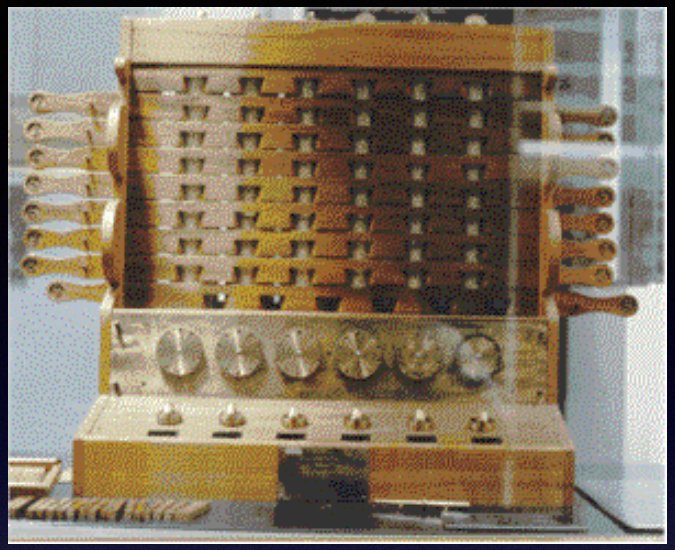
- Executa em qualquer coisa, controle total sobre a máquina
- Exige poucos recursos
- Impõe poucas restrições
- Sintaxe similar a outras linguagens, php, Java
- + Rigor e Disciplina
- Por outro lado: pouca flexibilidade, orientação a objetos (OO), …

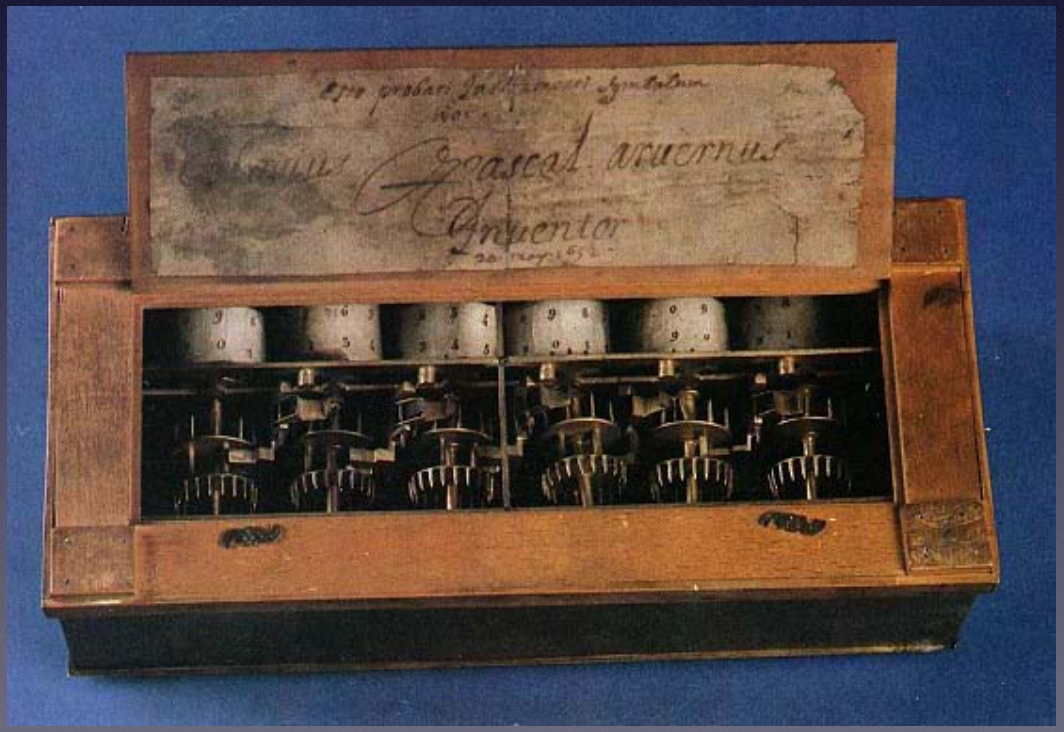
História





300 B.C.

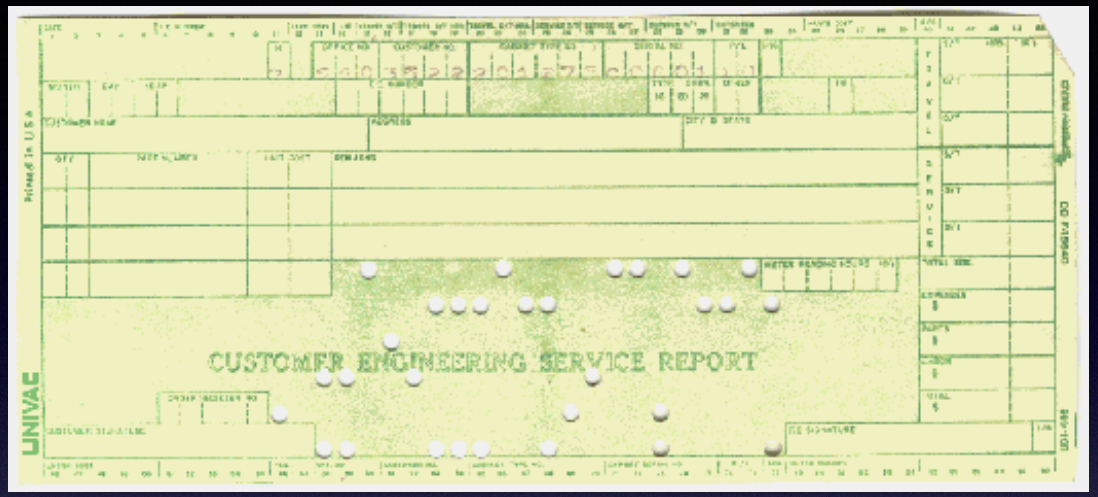


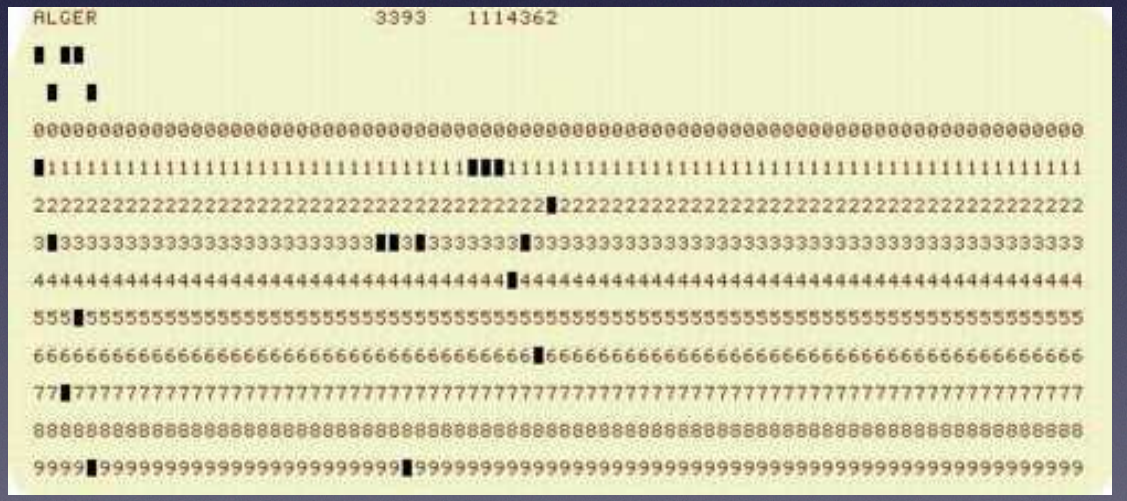


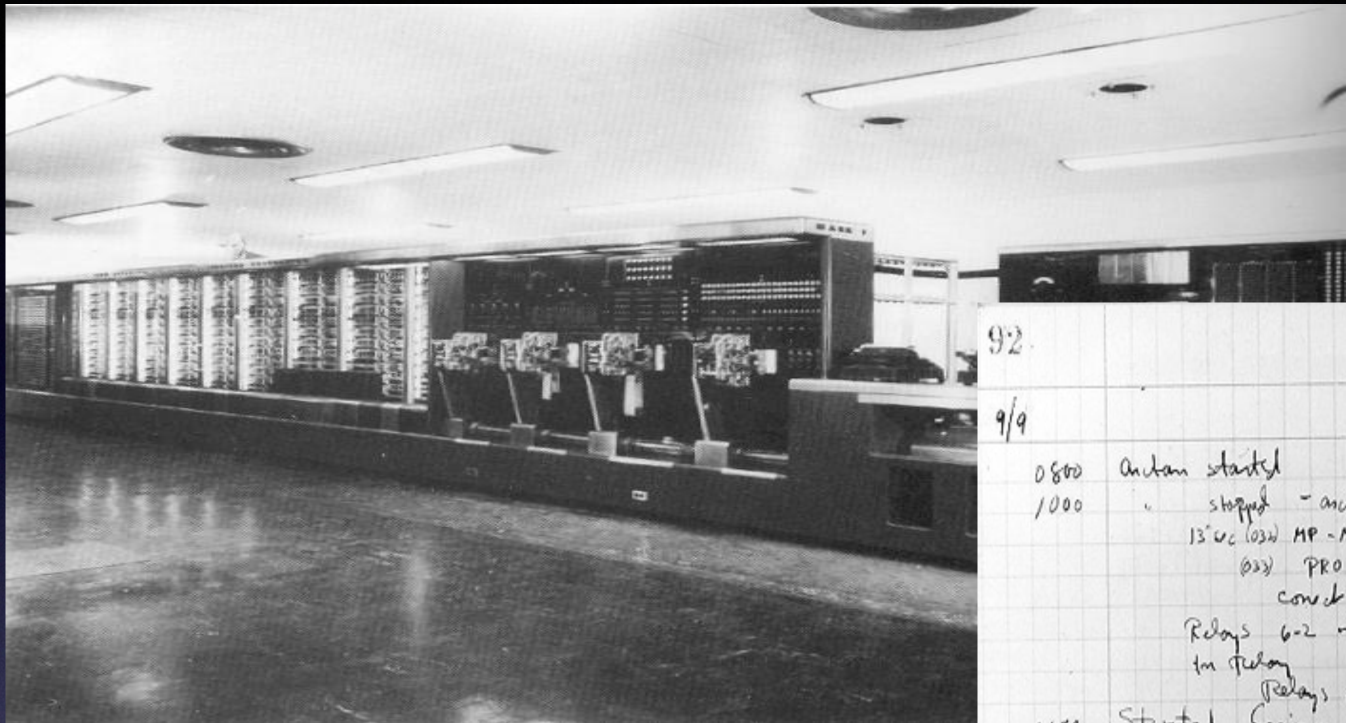
1642

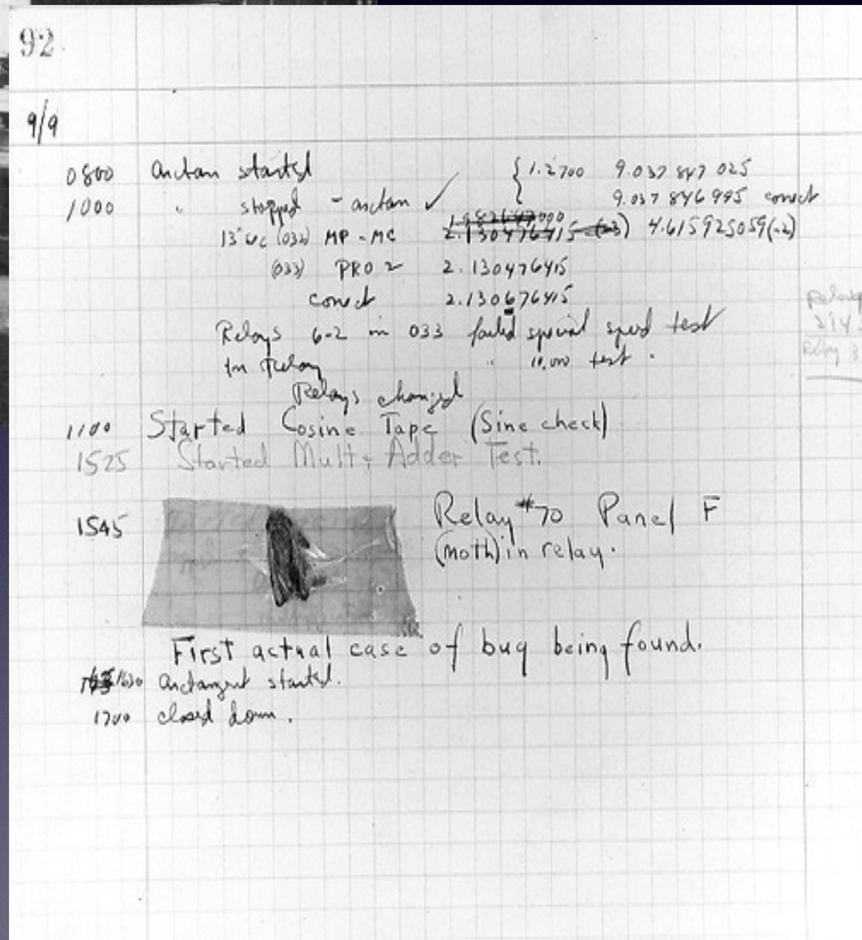
1822

1890

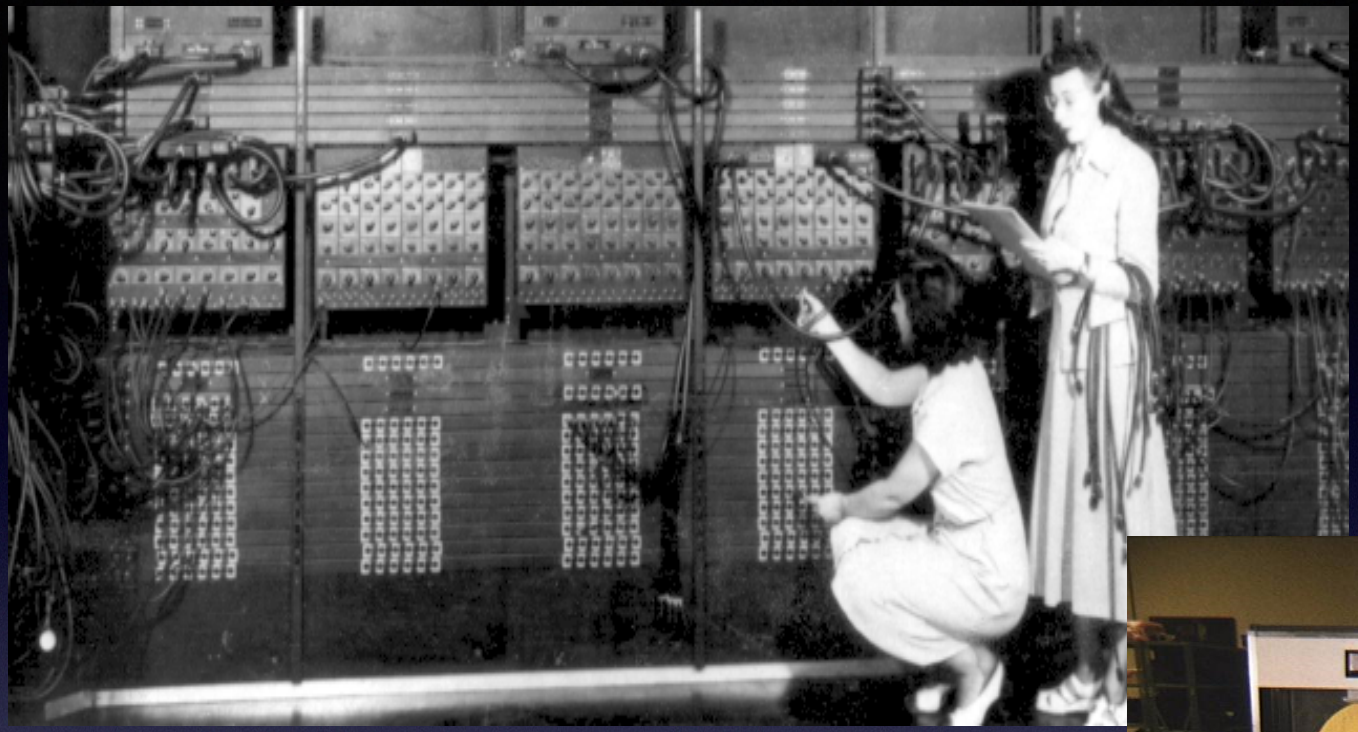

UNIVAC
CUSTOMER ENGINEERING SERVICE REPORT


ALGER
3393
1114362




92
9/9
0800 Antan started
1000 " stopped - antan ✓
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult+ Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1700 closed down.

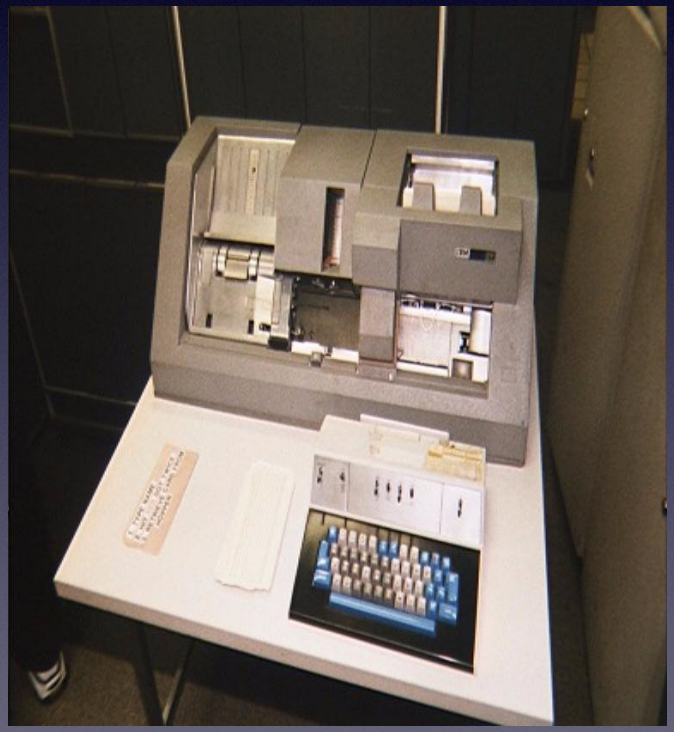
1944



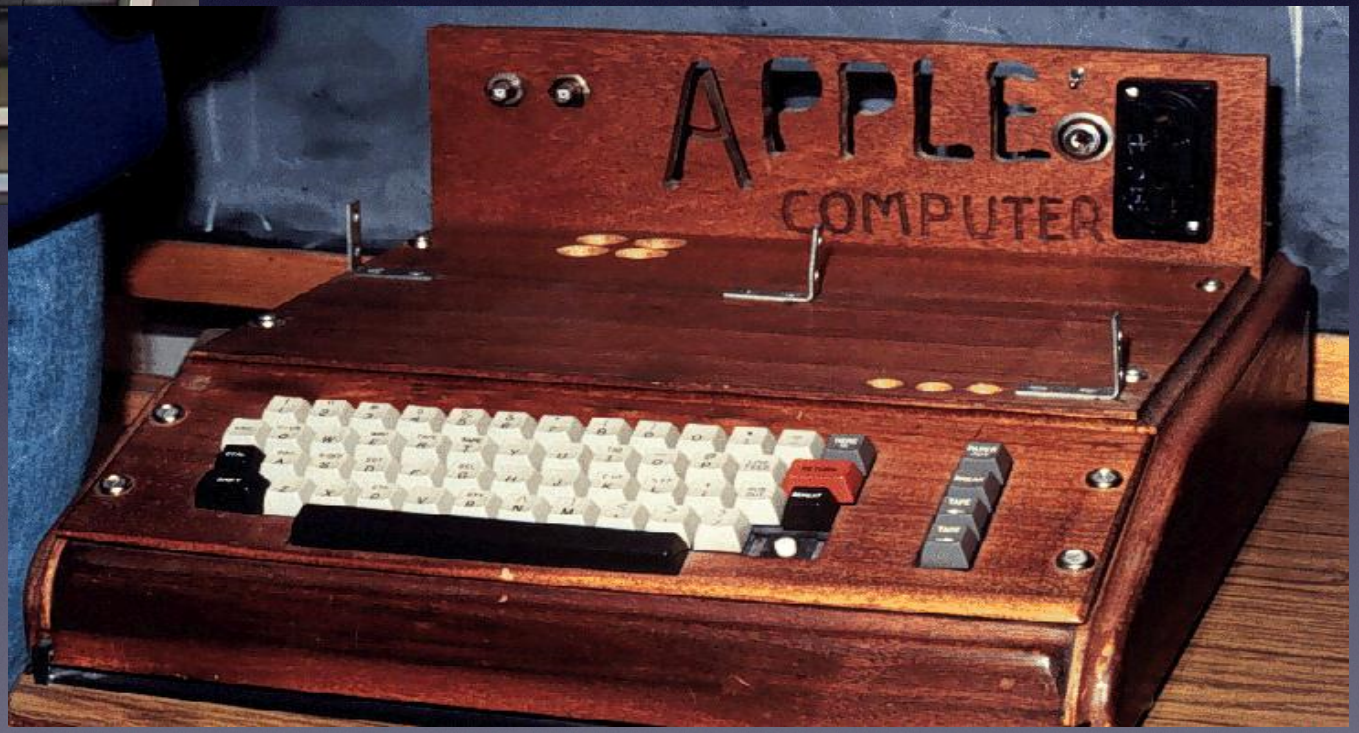
1945

Decada de 50

Decada de 60/70

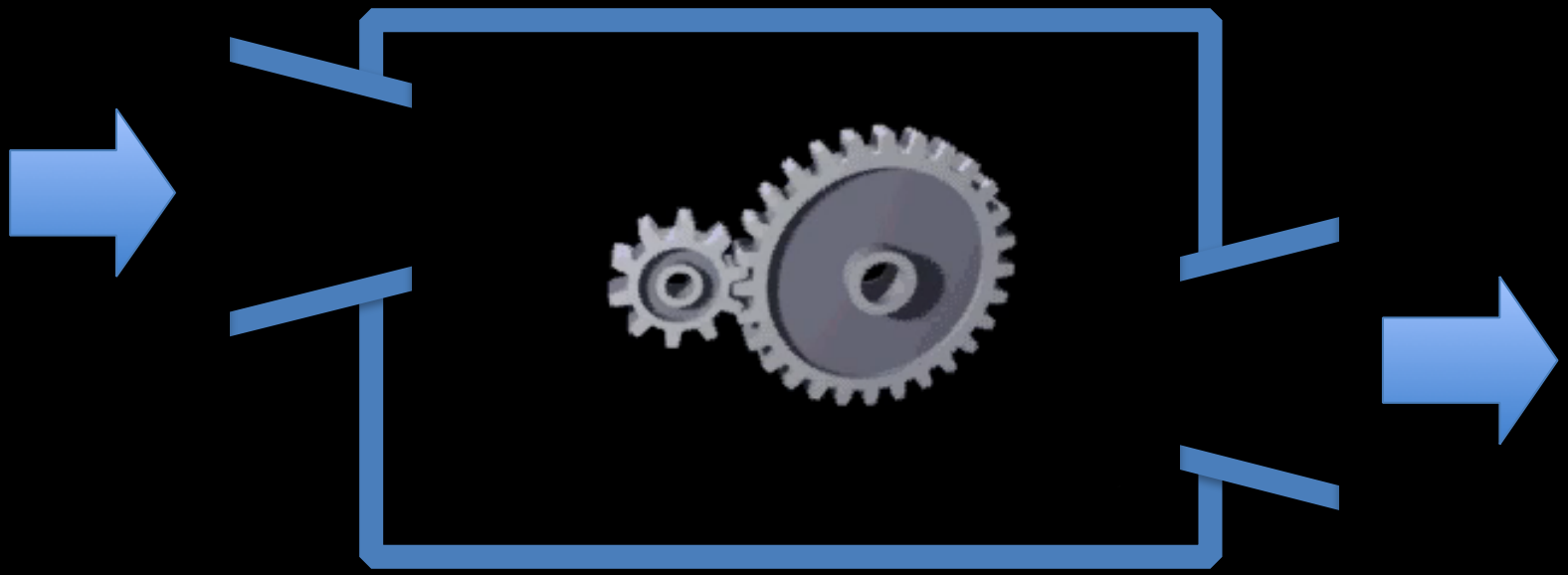


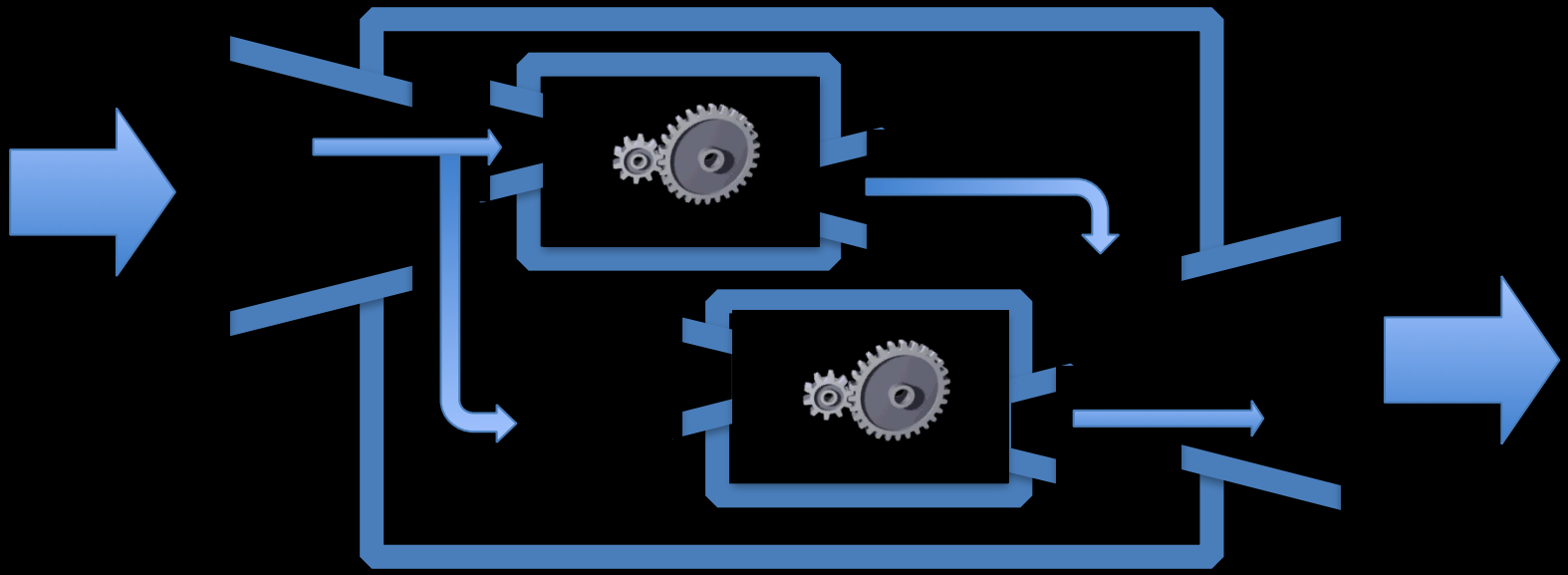
Decada de 70/80



Decada de 80/90

Algoritmos?





GIGO