

MAC 113 – Introdução à Ciência da Computação

Aula 16

Nelson Lago

1º/2025





E a prova, hein?

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

Questão 1

a: 38102289

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289

a': 89

x: 9

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289

a': 89

x: 9

y: 8

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

a: 38102

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

a: 38102 a': 2

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

a: 38102 a': 2 x: 2 y: 0

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

a: 38102 a': 2 x: 2 y: 0

saída: 2 * 10 + 0 = **20**

a: 3810 a': 10 x: 0 y: 1

saída: 0 * 10 + 1 = **1**

a: 381

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: 9 * 10 + 8 = **98**

a: 3810228 a': 28 x: 8 y: 2

saída: 8 * 10 + 2 = **82**

a: 381022 a': 22 x: 2 y: 2

saída: 2 * 10 + 2 = **22**

a: 38102 a': 2 x: 2 y: 0

saída: 2 * 10 + 0 = **20**

a: 3810 a': 10 x: 0 y: 1

saída: 0 * 10 + 1 = **1**

a: 381 a': 81 x: 1

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

a: 3

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

a: 3 a': 3

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

a: 3 a': 3 x: 3

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

a: 3 a': 3 x: 3 y: 0

Questão 1

```
main <- function() {  
  a <- 38102289  
  while (a > 0) {  
    f(a %% 100)  
    a <- a %/% 10  
  }  
}  
f <- function(a) {  
  x <- a %% 10  
  y <- a %/% 10  
  cat (x * 10 + y, "\n")  
}  
main()
```

a: 38102289 a': 89 x: 9 y: 8

saída: $9 * 10 + 8 = 98$

a: 3810228 a': 28 x: 8 y: 2

saída: $8 * 10 + 2 = 82$

a: 381022 a': 22 x: 2 y: 2

saída: $2 * 10 + 2 = 22$

a: 38102 a': 2 x: 2 y: 0

saída: $2 * 10 + 0 = 20$

a: 3810 a': 10 x: 0 y: 1

saída: $0 * 10 + 1 = 1$

a: 381 a': 81 x: 1 y: 8

saída: $1 * 10 + 8 = 18$

a: 38 a': 38 x: 8 y: 3

saída: $8 * 10 + 3 = 83$

a: 3 a': 3 x: 3 y: 0

saída: $3 * 10 + 0 = 30$

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]
```

L1

```
  viés <- runif(1, 1, 2)
```

L2

```
    scoreA <- scoreA + viés
```

L3

```
  } else {
```

```
    scoreB <- scoreB + viés
```

L4

```
  }
```

```
}
```

L5

L6

L7

```
  scoreA <- 5
```

```
  scoreB <- 5
```

```
}
```

L8

```
  return(a)
```

```
} else {
```

```
  return(b)
```

```
}
```

```
}
```

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]
```

L1

```
  viés <- runif(1, 1, 2)
```

L2

```
    scoreA <- scoreA + viés
```

```
    if (scoreA > 10) { scoreA <- 10 }
```

```
  } else {
```

```
    scoreB <- scoreB + viés
```

```
    if (scoreB > 10) { scoreB <- 10 }
```

```
  }
```

```
}
```

L5

L6

L7

```
  scoreA <- 5
```

```
  scoreB <- 5
```

```
}
```

L8

```
  return(a)
```

```
} else {
```

```
  return(b)
```

```
}
```

```
}
```

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]
```

L1

```
  viés <- runif(1, 1, 2)
```

L2

```
    scoreA <- scoreA + viés
```

```
    if (scoreA > 10) { scoreA <- 10 }
```

```
  } else {
```

```
    scoreB <- scoreB + viés
```

```
    if (scoreB > 10) { scoreB <- 10 }
```

```
  }
```

```
}
```

L5

L6

L7

```
  scoreA <- 5
```

```
  scoreB <- 5
```

```
}
```

L8

```
  return(a)
```

```
} else {
```

```
  return(b)
```

```
}
```

```
}
```

L1:

if (!honesto) {

if (honesto) {

if (runif(1, 0, 1) >= 0.2) {

if (scoreA > scoreB) {

if (runif(1, 0, 1) <= 0.2) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]
```

L1

```
  viés <- runif(1, 1, 2)
```

L2

```
    scoreA <- scoreA + viés  
    if (scoreA > 10) { scoreA <- 10 }  
  } else {  
    scoreB <- scoreB + viés  
    if (scoreB > 10) { scoreB <- 10 }  
  }  
}
```

L5

L6

L7

```
  scoreA <- 5
```

```
  scoreB <- 5
```

```
}
```

L8

```
  return(a)
```

```
} else {
```

```
  return(b)
```

```
}
```

```
}
```

L1:

if (!honesto) {

if (honesto) {

if (runif(1, 0, 1) >= 0.2) {

if (scoreA > scoreB) {

if (runif(1, 0, 1) <= 0.2) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    L2  
    scoreA <- scoreA + viés  
    if (scoreA > 10) { scoreA <- 10 }  
  } else {  
    scoreB <- scoreB + viés  
    if (scoreB > 10) { scoreB <- 10 }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```


Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    L2  
    scoreA <- scoreA + viés  
    if (scoreA > 10) { scoreA <- 10 }  
  } else {  
    scoreB <- scoreB + viés  
    if (scoreB > 10) { scoreB <- 10 }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

L2:

if (runif(1, 1, 2) >= 0.5) {

if (runif(1, 0, 1) >= 0.5) {

if (scoreA > scoreB) {

if (runif(1, 0, 1) >= 0.8) {

if (scoreA < scoreB) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    L2  
    scoreA <- scoreA + viés  
    if (scoreA > 10) { scoreA <- 10 }  
  } else {  
    scoreB <- scoreB + viés  
    if (scoreB > 10) { scoreB <- 10 }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

L2:

if (runif(1, 1, 2) >= 0.5) {



if (runif(1, 0, 1) >= 0.5) {

if (scoreA > scoreB) {

if (runif(1, 0, 1) >= 0.8) {

if (scoreA < scoreB) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

L7: if ((scoreA - scoreB) / d >= 0.5) {
 if ((scoreA - scoreB) / (scoreA + scoreB) >= 0.5) {
 if ((scoreA - scoreB) / (scoreA + scoreB) <= 0.5) {

if (runif(1, 1, 2) >= d / (scoreA + scoreB)) {
 if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
L5  
L6  
L7  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

- L7:
- if ((scoreA - scoreB) / d >= 0.5) {
 - if ((scoreA - scoreB) / (scoreA + scoreB) >= 0.5) {
 - if ((scoreA - scoreB) / (scoreA + scoreB) <= 0.5) {

- if (runif(1, 1, 2) >= d / (scoreA + scoreB)) {
- if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

L5

L6

```
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}
```

L8

```
  return(a)  
} else {  
  return(b)  
}
```

}

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

L5

L6

```
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}
```

L8

```
  return(a)  
} else {  
  return(b)  
}  
}
```

L5:

- $d <- \text{scoreA} - \text{scoreB}$
 $d <- \text{scoreA} / \text{scoreB}$
 $d <- \text{scoreA} + \text{scoreB}$

- $d <- (\text{scoreA} - \text{scoreB}) / (\text{scoreA} + \text{scoreB})$
 $d <- 1 / (\text{scoreA} + \text{scoreB})$

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

L5

L6

```
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}
```

L8

```
return(a)  
} else {  
  return(b)  
}  
}
```

L5:



$d <- \text{scoreA} - \text{scoreB}$



$d <- \text{scoreA} / \text{scoreB}$



$d <- \text{scoreA} + \text{scoreB}$



$d <- (\text{scoreA} - \text{scoreB}) / (\text{scoreA} + \text{scoreB})$



$d <- 1 / (\text{scoreA} + \text{scoreB})$

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
L6  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
L6  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

L6:

if (d < 0) { d <- d * -1 }

if (d <= 0) { d <- 1 }

while (d < 0) { d <- d + 1 }

if (d <= 0) { d <- scoreA }

if (d > 0) { d <- d * -1 }

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB
```

L6

```
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}
```

L8

```
return(a)  
} else {  
  return(b)  
}
```

```
}
```

L6:



if (d < 0) { d <- d * -1 }



if (d <= 0) { d <- 1 }



while (d < 0) { d <- d + 1 }



if (d <= 0) { d <- scoreA }



if (d > 0) { d <- d * -1 }

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
if (d < 0) { d <- d * -1 }  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
  
  return(a)  
} else {  
  return(b)  
}  
}
```

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
if (d < 0) { d <- d * -1 }  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

- L8:
- | | | | |
|--------------------------|---|--------------------------|---|
| <input type="checkbox"/> | if (runif(1, 1, 2) <= scoreA / (scoreA + scoreB)) { | <input type="checkbox"/> | if (scoreA > scoreB) { |
| <input type="checkbox"/> | if (runif(1, 0, 1) >= scoreA / (scoreA + scoreB)) { | <input type="checkbox"/> | if (runif(1, 0, 1) <= scoreA / (scoreA + scoreB)) { |
| <input type="checkbox"/> | if (scoreA / (scoreA + scoreB) >= 0.5) { | | |

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
if (d < 0) { d <- d * -1 }  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
L8  
  return(a)  
} else {  
  return(b)  
}  
}
```

L8:

- if (runif(1, 1, 2) <= scoreA / (scoreA + scoreB)) {
 if (runif(1, 0, 1) >= scoreA / (scoreA + scoreB)) {
 if (scoreA / (scoreA + scoreB) >= 0.5) {

- if (scoreA > scoreB) {
 if (runif(1, 0, 1) <= scoreA / (scoreA + scoreB)) {

Questão 2

```
partida <- function(a, b, scores) {  
  scoreA <- scores[a]  
  scoreB <- scores[b]  
  
  if (runif(1, 0, 1) <= 0.2) {  
    viés <- runif(1, 1, 2)  
    if (runif(1, 0, 1) >= 0.5) {  
      scoreA <- scoreA + viés  
      if (scoreA > 10) { scoreA <- 10 }  
    } else {  
      scoreB <- scoreB + viés  
      if (scoreB > 10) { scoreB <- 10 }  
    }  
  }  
}
```

```
d <- scoreA - scoreB  
if (d < 0) { d <- d * -1 }  
if (runif(1, 0, 1) >= d / (scoreA + scoreB)) {  
  scoreA <- 5  
  scoreB <- 5  
}  
  
if (runif(1, 0, 1) <= scoreA / (scoreA + scoreB))  
  return(a)  
} else {  
  return(b)  
}  
}
```

Questão 3 – 1/2

média: $\frac{\sum_{i=1}^n \text{val}_i}{n}$

variância: $\frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$

desvio padrão: $\sqrt{\text{variância}}$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    L1  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```


Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    L1  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

- L1:
- | | | | |
|--------------------------|-------------------------------------|--------------------------|----------------------------|
| <input type="checkbox"/> | média <- (média + v[i]) / length(v) | <input type="checkbox"/> | média <- (média + v[i])**2 |
| <input type="checkbox"/> | média <- sqrt(média + v[i]) | <input type="checkbox"/> | média <- média + v[i] |
| <input type="checkbox"/> | média <- v[i] / length(v) | | |

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    L1  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

- L1:
- média <- (média + v[i]) / length(v)
 - média <- sqrt(média + v[i])
 - média <- v[i] / length(v)

- média <- (média + v[i])**2
- média <- média + v[i]

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

- L2:
- | | | | |
|--------------------------|--|--------------------------|--|
| <input type="checkbox"/> | variância <- (v[i] - média)**2 | <input type="checkbox"/> | variância <- variância + (v[i] - média)**2 |
| <input type="checkbox"/> | variância <- (variância + (v[i] - média)**2 | <input type="checkbox"/> | variância <- (variância + v[i] - média)**2 |
| <input type="checkbox"/> | variância <- ((variância + v[i] + média) / 500)**2 | | |

Questão 3 – 1/2

média: $\frac{\sum_{i=1}^n \text{val}_i}{n}$

variância: $\frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$

desvio padrão: $\sqrt{\text{variância}}$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  L2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

- L2:
- variância <- (v[i] - média)**2
 - variância <- variância + (v[i] - média)**2
 - variância <- (variância + (v[i] - média))**2
 - variância <- ((variância + v[i] + média) / 500)**2

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  variância <- variância + (v[i] - média)**2  
  i <- i + 1  
}  
variância <- variância / length(v)  
  
}
```

Questão 3 – 1/2

média: $\frac{\sum_{i=1}^n \text{val}_i}{n}$

variância: $\frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$

desvio padrão: $\sqrt{\text{variância}}$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  variância <- variância + (v[i] - média)**2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

L3:

return (variância)

return (sqrt(variância / length(v)))

return (sqrt(variância) / length(v))

return (sqrt(variância**2 / length(v)**2))

return (sqrt(variância))

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  variância <- variância + (v[i] - média)**2  
  i <- i + 1  
}  
variância <- variância / length(v)  
L3  
}
```

- L3:
- return (variância)
 - return (sqrt(variância / length(v)))
 - return (sqrt(variância) / length(v))

- return (sqrt(variância**2 / length(v)**2))
- return (sqrt(variância))

Questão 3 – 1/2

$$\text{média: } \frac{\sum_{i=1}^n \text{val}_i}{n}$$

$$\text{variância: } \frac{\sum_{i=1}^n (\text{val}_i - \text{média})^2}{n}$$

$$\text{desvio padrão: } \sqrt{\text{variância}}$$

```
calcula_desvio <- function(v) {  
  média <- 0  
  i <- 1  
  while (i <= length(v)) {  
    média <- média + v[i]  
    i <- i + 1  
  }  
  média <- média / length(v)  
}
```

```
variância <- 0  
i <- 1  
while (i <= length(v)) {  
  variância <- variância + (v[i] - média)**2  
  i <- i + 1  
}  
variância <- variância / length(v)  
return (sqrt(variância))  
}
```

Questão 3 – 2/2

```
main <- function() {  
  dentro <- 0  
  estimativas <- numeric(500)  
  desvio <- 1  
  iterações <- 1  
  ...  
  cat(glue("Após {iteraões} iteraões, estimamos que pi é cerca de {4*dentro/iteraões}"),  
    }  
  main()  
}
```

Questão 3 – 2/2

```
i <- 1
L4
  L5
  L6
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7
    desvio <- calcula_desvio(estimativas)
    L8
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

- L4: while (desvio > 0) { while (desvio < 500) {
 while (desvio < 10⁻⁶) { while (desvio > 10⁻⁶) {
 while (4 * dentro / iterações - 3.141592 > 10⁻⁶) {

Questão 3 – 2/2

```
i <- 1
L4
  L5
  L6
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7
    desvio <- calcula_desvio(estimativas)
    L8
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

- L4:
- | | | | |
|--------------------------|---|-------------------------------------|--------------------------------------|
| <input type="checkbox"/> | while (desvio > 0) { | <input type="checkbox"/> | while (desvio < 500) { |
| <input type="checkbox"/> | while (desvio < 10 ⁻⁶) { | <input checked="" type="checkbox"/> | while (desvio > 10 ⁻⁶) { |
| <input type="checkbox"/> | while (4 * dentro / iterações - 3.141592 > 10 ⁻⁶) { | | |

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  L5
  L6
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7
  desvio <- calcula_desvio(estimativas)
  L8
}
iterações <- iterações + 1
i <- i + 1
}
```

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  L5
  L6
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7
  desvio <- calcula_desvio(estimativas)
  L8
}
iteraões <- iteraões + 1
i <- i + 1
}
```

L5: x <- runif(-1, 1)
 x <- runif(1, -1, 1)

x <- runif(1)
 x <- runif(1, 1, 2)

x <- runif(1, 0, 1)

L6: y <- runif(1)
 y <- runif(-1, 1)

y <- runif(1, 0, 1)
 y <- runif(1, 1, 2)

y <- runif(1, -1, 1)

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  L5
  L6
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7
  desvio <- calcula_desvio(estimativas)
  L8
}
iteraões <- iteraões + 1
i <- i + 1
}
```

L5: x <- runif(-1, 1)
 x <- runif(1, -1, 1)

x <- runif(1)
 x <- runif(1, 1, 2)

x <- runif(1, 0, 1)

L6: y <- runif(1)
 y <- runif(-1, 1)

y <- runif(1, 0, 1)
 y <- runif(1, 1, 2)

y <- runif(1, -1, 1)

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7 
  desvio <- calcula_desvio(estimativas)
  L8 
}
iteraões <- iteraões + 1
i <- i + 1
}
```


Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7 
  desvio <- calcula_desvio(estimativas)
  L8 
}
iteraões <- iteraões + 1
i <- i + 1
}
```

L7:

while (i > 0) {

if (i == 500) {

if (i > 500) {

while (i <= 500) {

while (i < 500) {

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  L7 
  desvio <- calcula_desvio(estimativas)
  L8 
}
iteraões <- iteraões + 1
i <- i + 1
}
```

L7:

while (i > 0) {



if (i == 500) {

if (i > 500) {

while (i <= 500) {

while (i < 500) {

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  if (i == 500) {
    desvio <- calcula_desvio(estimativas)
    L8
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  if (i == 500) {
    desvio <- calcula_desvio(estimativas)
    L8
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

L8:

`i <- i + 1`

`i <- 0`

`i <- 500`

`i <- i - 1`

`i <- 1`

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  if (i == 500) {
    desvio <- calcula_desvio(estimativas)
    L8
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

L8: i <- i + 1
 i <- i - 1

i <- 0
 i <- 1

i <- 500

Questão 3 – 2/2

```
i <- 1
while (desvio > 10^-6) {
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)
  if (x**2 + y**2 <= 1) { dentro <- dentro + 1 }
  estimativas[i] <- 4 * dentro / iterações
  if (i == 500) {
    desvio <- calcula_desvio(estimativas)
    i <- 0
  }
  iterações <- iterações + 1
  i <- i + 1
}
```

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “previous(**var**)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

Questão 4

- A condição para a execução ou não de um bloco **if** é colocada entre parênteses
- É sempre preferível usar nomes simples, como a, b, x, y etc., para as variáveis de um programa
- Um único *vector* pode armazenar vários tipos de dado diferentes
- “x <- **integer**(4)” armazena o inteiro 4 na variável x
- Os operadores “&&”, “!” e “||” são variáveis do tipo LOGICAL
- Uma função definida como f <- **function**(a, b){...} precisa usar **readline**() para obter os valores de a e b
- Após uma atribuição a uma variável preexistente **var**, é possível obter o valor anterior dessa variável com “**previous**(var)”
- A operação representada por == sempre resulta em um valor do tipo LOGICAL
- O método de Monte Carlo permite obter soluções aproximadas através da análise estatística de várias simulações
- A variável de controle de um laço **while** só pode mudar de valor zero ou uma vez
- Um indicador de passagem sinaliza se uma determinada condição aconteceu ou não durante a execução do código ou laço
- Em um laço **while**, a condição para a continuidade ou o término das repetições geralmente depende de uma variável

And now for something completely different

Doncovim?!

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um —

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um – |

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um – ||

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um – |||

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**

- ▶ Relação um para um – **||||**

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um – |||||...

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**
 - ▶ Relação um para um – |||||...
- **Escrita e algarismos**

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Contar nos dedos, contar “palitinhos”...**

- ▶ Relação um para um – |||||...

- **Escrita e algarismos**

- ▶ 4 operações, grandes números, “vai um” (“carry”)

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

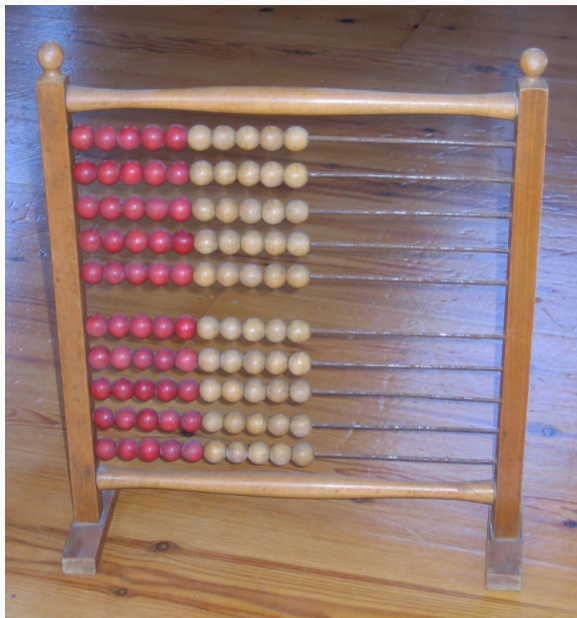
- **Contar nos dedos, contar “palitinhos”...**

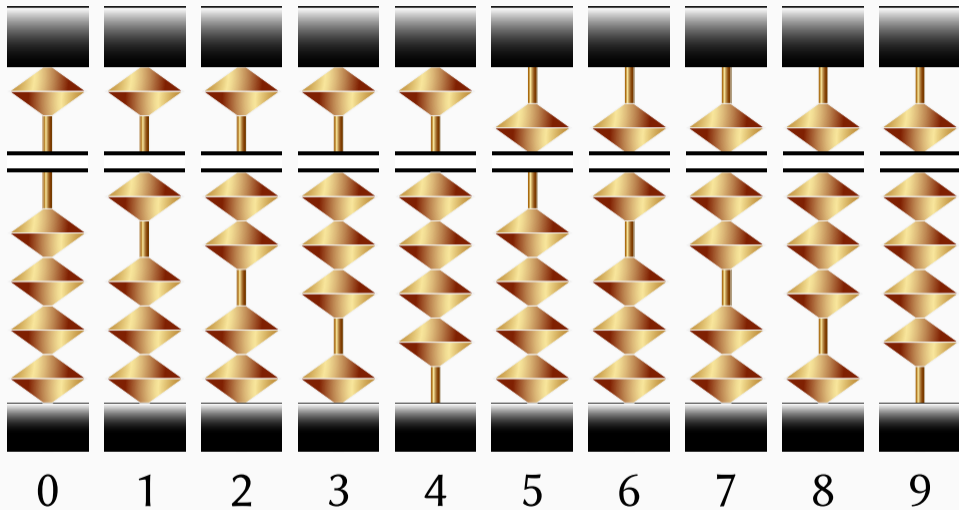
- ▶ Relação um para um – |||||...

- **Escrita e algarismos**

- ▶ 4 operações, grandes números, “vai um” (“carry”)

- ▶ Ábaco





Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Réguas, balanças**

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Réguas, balanças**
- **Relógios de sol e clepsidras**

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Astrolábio (séc II AC)**

- ▶ Permite determinar sua posição no solo observando as estrelas (entre outras funções)

Até o início do século XX, a história da computação é, em grande parte, um capítulo na história da matemática

- **Astrolábio (séc II AC)**

- ▶ Permite determinar sua posição no solo observando as estrelas (entre outras funções)

- **Mecanismo de Anticítera (séc II AC)**

- ▶ Considerado o primeiro “computador analógico”, permite prever as posições dos astros e eclipses (ou seja, é mais que apenas um instrumento de medida)



[commons.wikimedia.org/wiki/File:Antikythera_mechanism_clockface,_1st-2nd_century_BC,_Greece_\(model\).jpg](https://commons.wikimedia.org/wiki/File:Antikythera_mechanism_clockface,_1st-2nd_century_BC,_Greece_(model).jpg)
pt.wikipedia.org/wiki/Ficheiro:Antikythera_model_front_panel_Mogi_Vicentini_2007.JPG

- Tabelas trigonométricas (séc I–VI)

- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)

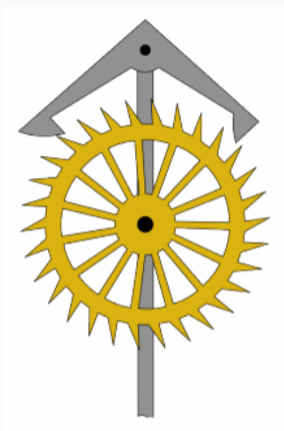
- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)
- **Números arábicos (na Europa, séc IX?)**
 - ▶ (muito mais fácil de automatizar)

- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)
- **Números arábicos (na Europa, séc IX?)**
 - ▶ (muito mais fácil de automatizar)
- **“Órgão” automático (na verdade, flauta – séc IX), precursor da pianola (séc XIX)**
 - ▶ muslimheritage.com/hydraulic-organ-of-banu-musa/
 - ▶ (um cilindro giratório com pinos abria e fechava os furos no corpo da plauta, como uma caixa de música moderna)

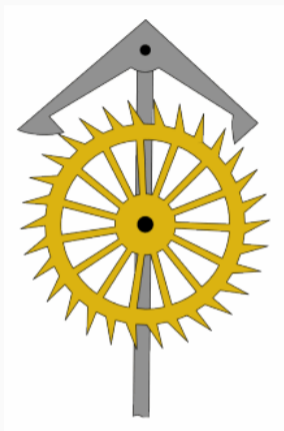
- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)
- **Números arábicos (na Europa, séc IX?)**
 - ▶ (muito mais fácil de automatizar)
- **“Órgão” automático (na verdade, flauta – séc IX), precursor da pianola (séc XIX)**
 - ▶ muslimheritage.com/hydraulic-organ-of-banu-musa/
 - ▶ (um cilindro giratório com pinos abria e fechava os furos no corpo da flauta, como uma caixa de música moderna)
 - ▶ Provavelmente o primeiro sistema “programável”, ou seja, em que há uma separação entre “hardware” e “software”

- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)
- **Números arábicos (na Europa, séc IX?)**
 - ▶ (muito mais fácil de automatizar)
- **“Órgão” automático (na verdade, flauta – séc IX), precursor da pianola (séc XIX)**
 - ▶ muslimheritage.com/hydraulic-organ-of-banu-musa/
 - ▶ (um cilindro giratório com pinos abria e fechava os furos no corpo da flauta, como uma caixa de música moderna)
 - ▶ Provavelmente o primeiro sistema “programável”, ou seja, em que há uma separação entre “hardware” e “software”
- **Relógios mecânicos (séc XIII–XIV)**

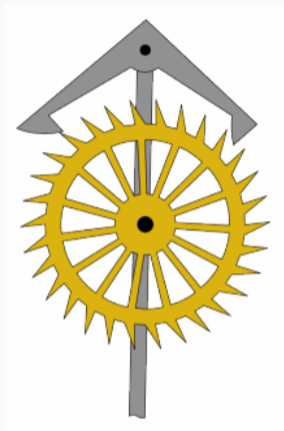
- **Tabelas trigonométricas (séc I–VI)**
 - ▶ Maior precisão com séries de Madhava (séc XIV)
- **Números arábicos (na Europa, séc IX?)**
 - ▶ (muito mais fácil de automatizar)
- **“Órgão” automático (na verdade, flauta – séc IX), precursor da pianola (séc XIX)**
 - ▶ muslimheritage.com/hydraulic-organ-of-banu-musa/
 - ▶ (um cilindro giratório com pinos abria e fechava os furos no corpo da flauta, como uma caixa de música moderna)
 - ▶ Provavelmente o primeiro sistema “programável”, ou seja, em que há uma separação entre “hardware” e “software”
- **Relógios mecânicos (séc XIII–XIV)**
 - ▶ Início dos mecanismos de precisão



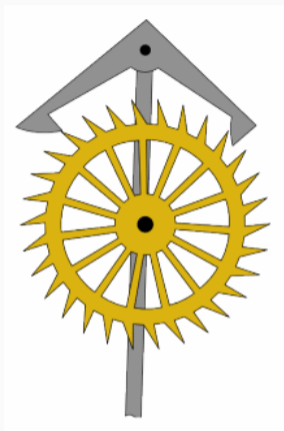
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



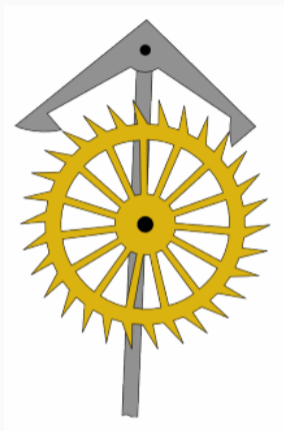
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



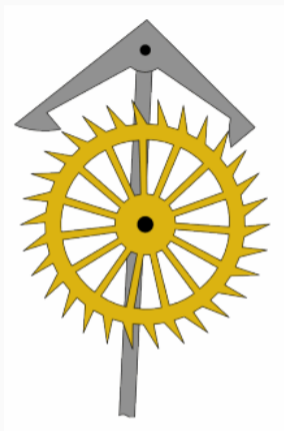
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



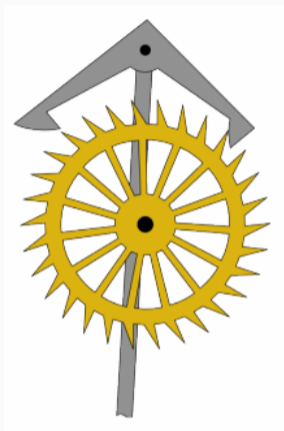
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



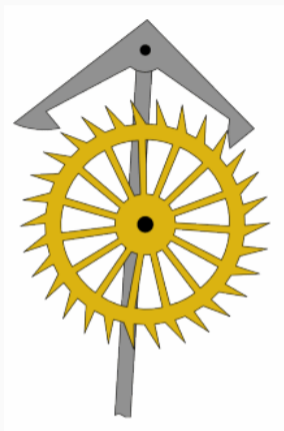
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



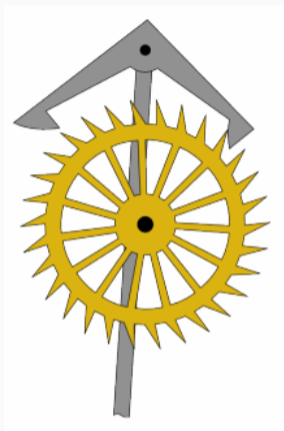
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



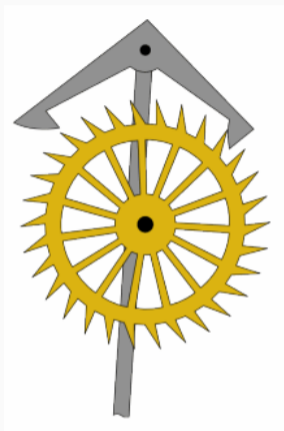
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



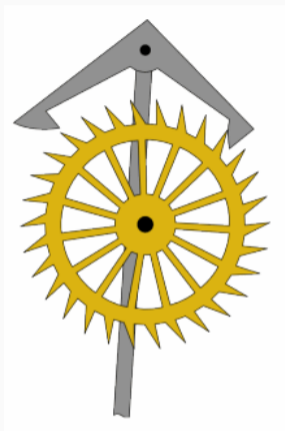
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



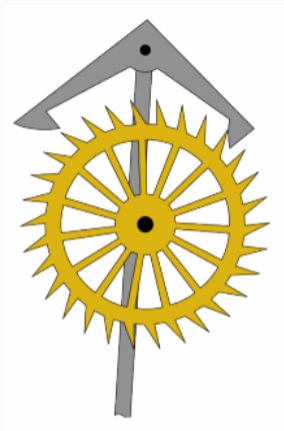
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



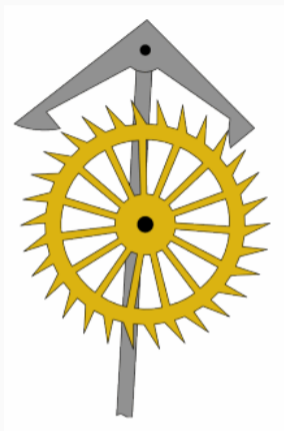
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



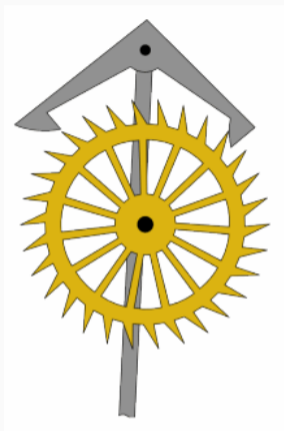
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



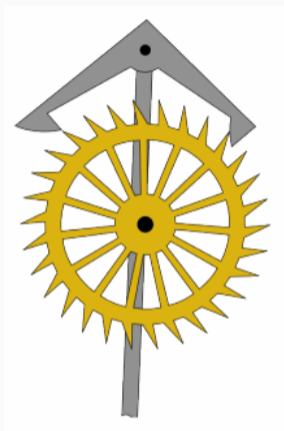
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



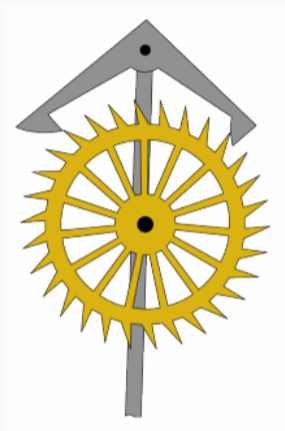
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



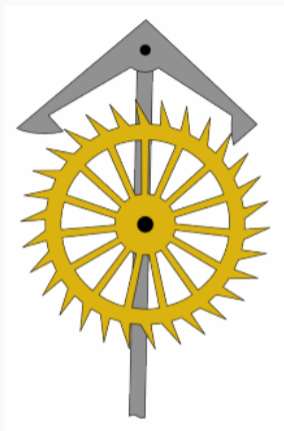
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



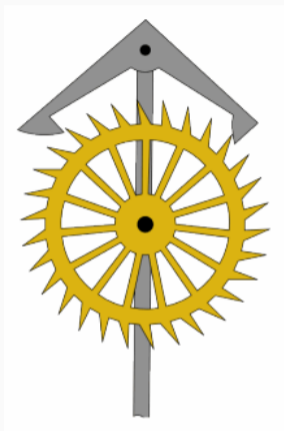
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



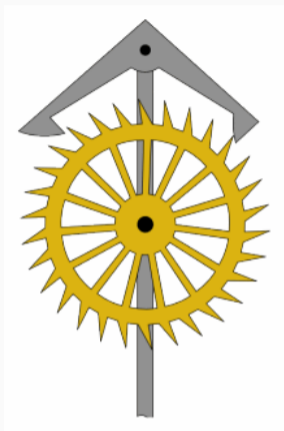
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



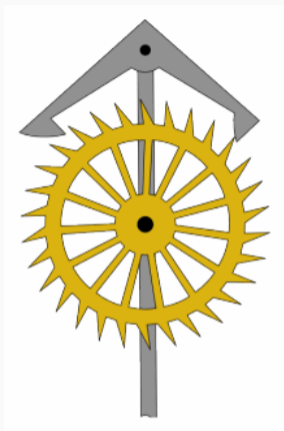
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



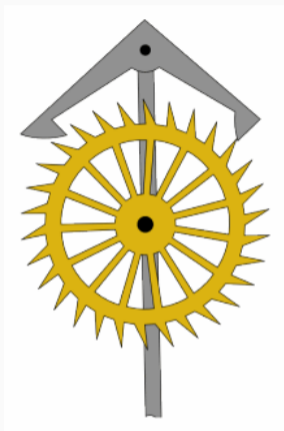
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



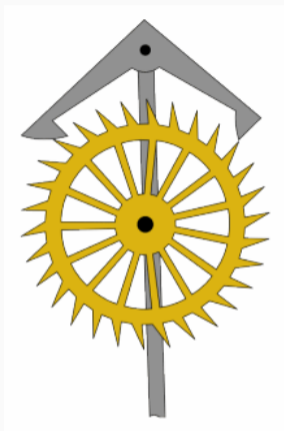
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



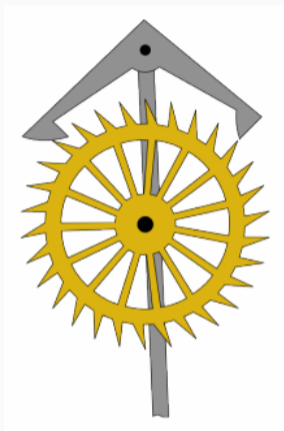
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



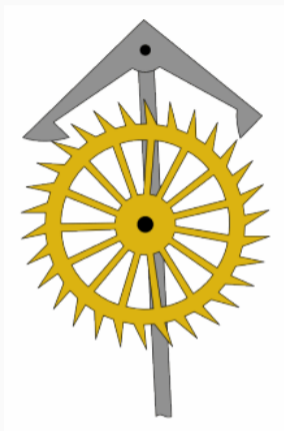
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



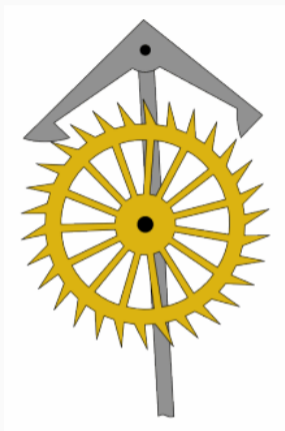
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



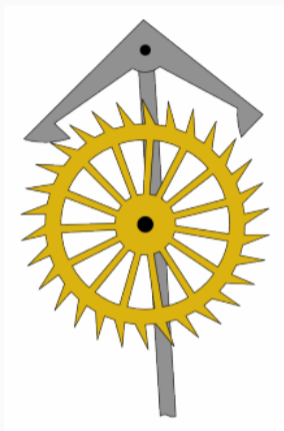
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



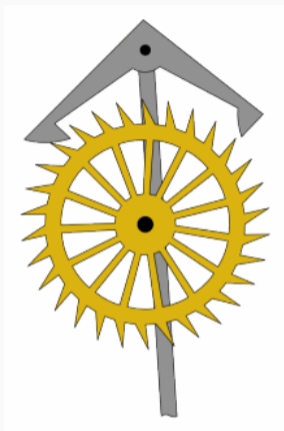
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



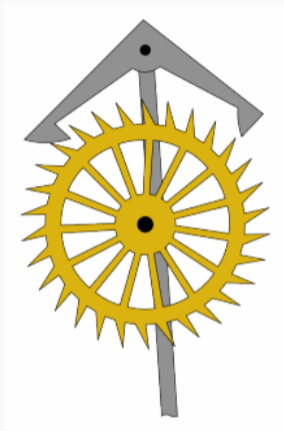
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



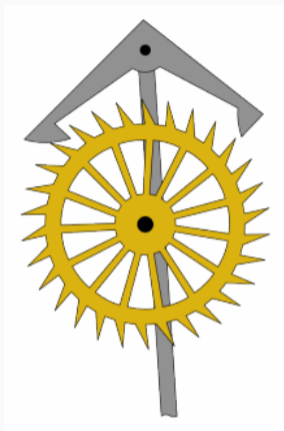
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



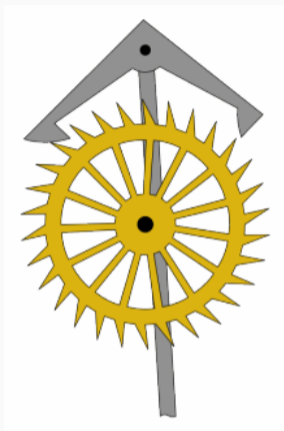
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



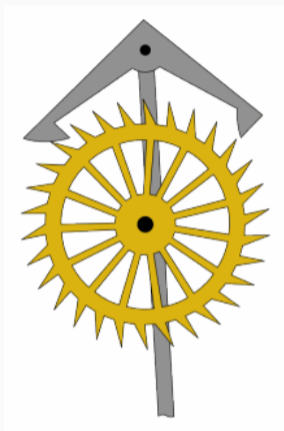
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



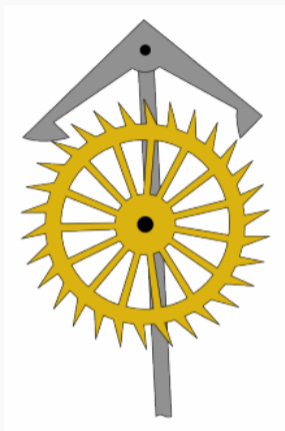
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



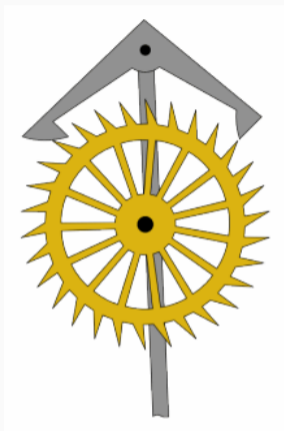
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



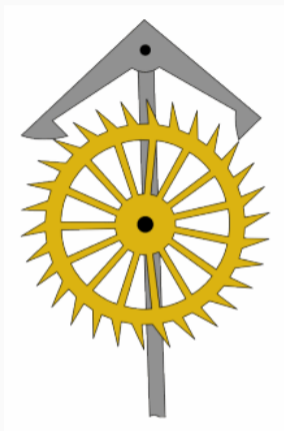
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



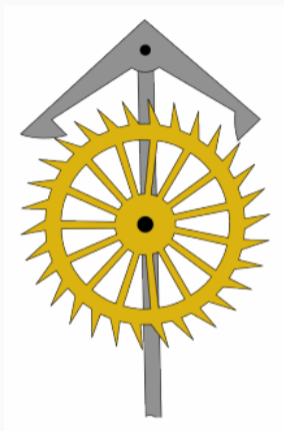
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



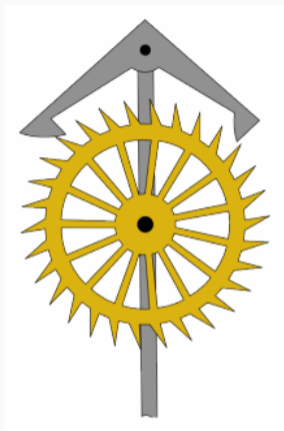
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



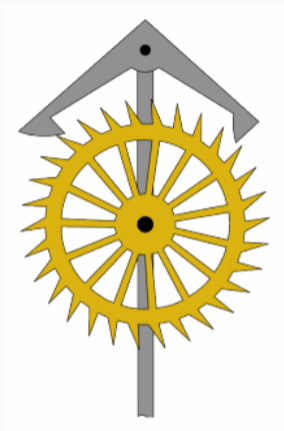
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



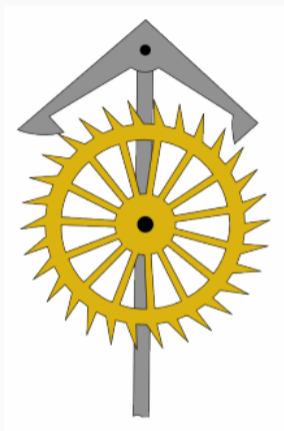
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



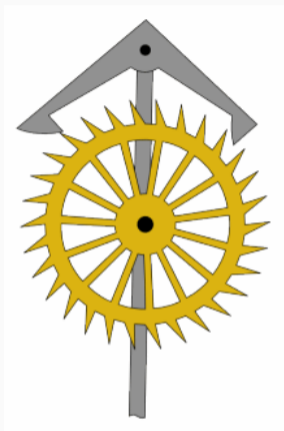
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



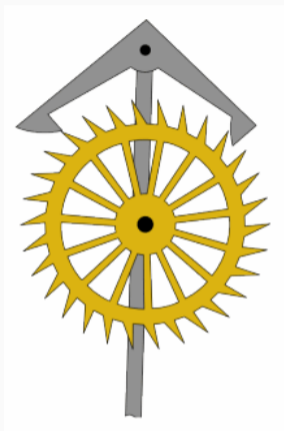
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



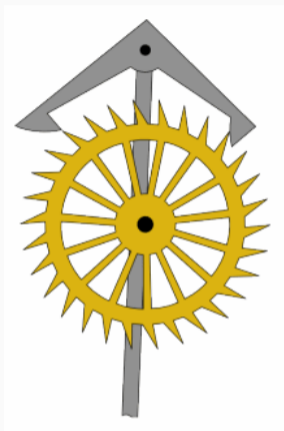
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



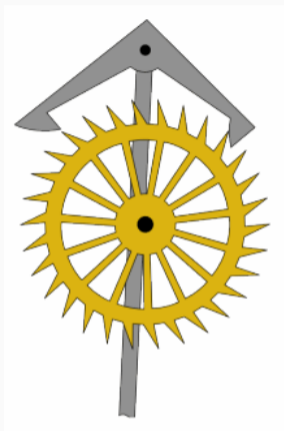
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



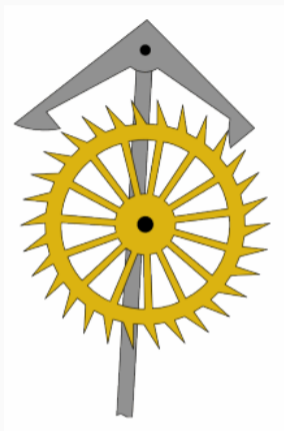
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



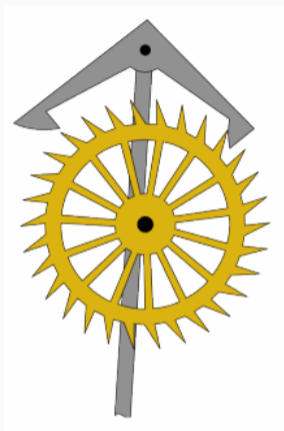
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



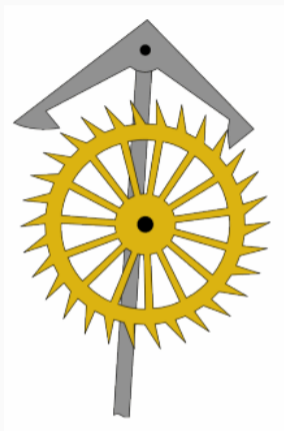
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



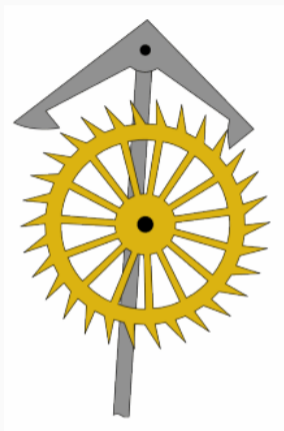
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



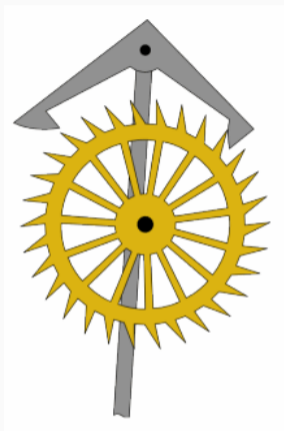
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



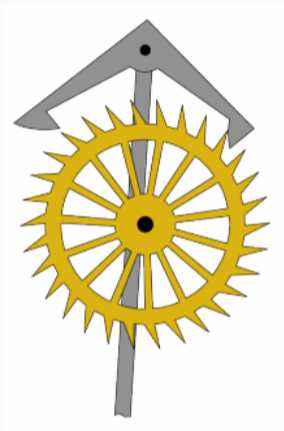
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



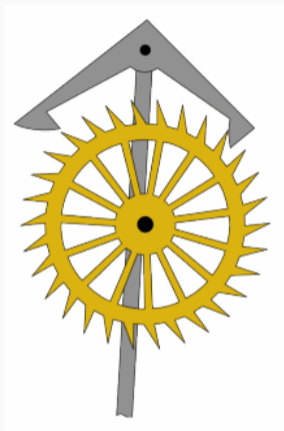
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



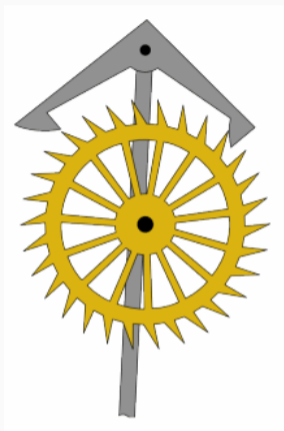
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



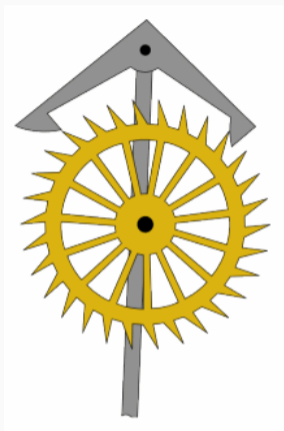
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



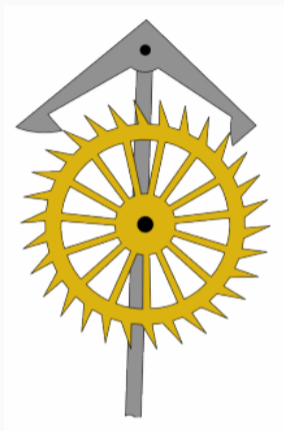
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



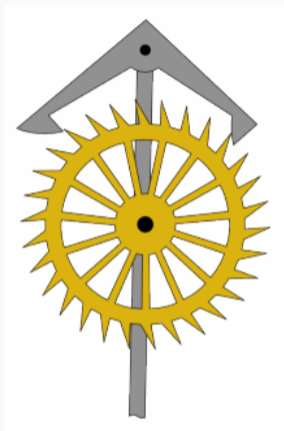
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



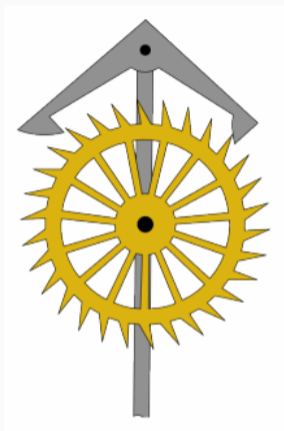
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



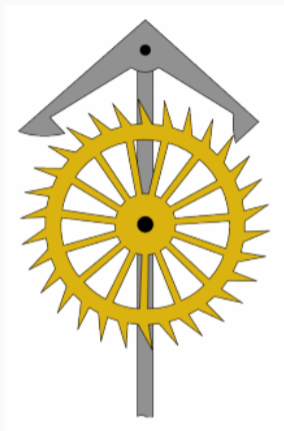
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



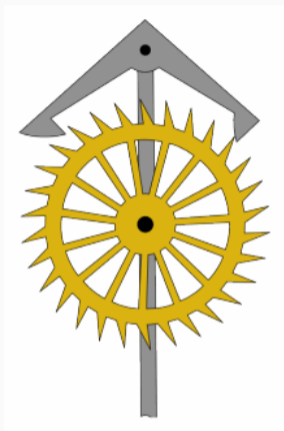
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



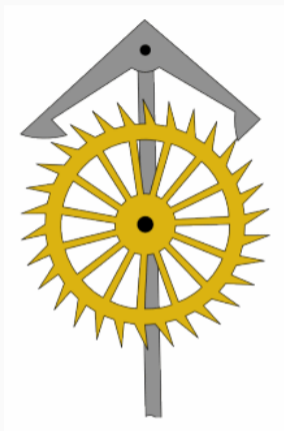
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



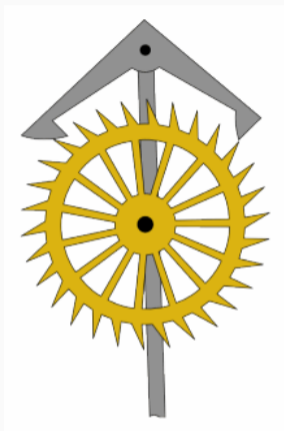
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



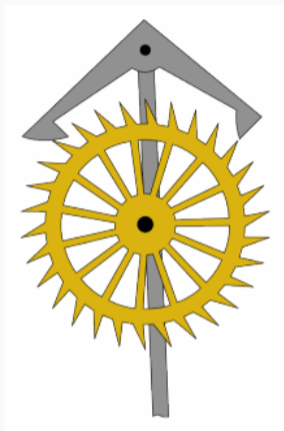
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



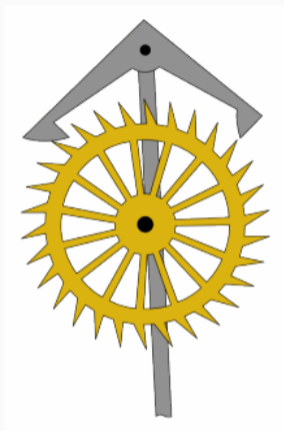
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



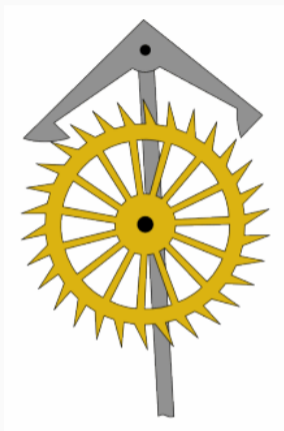
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



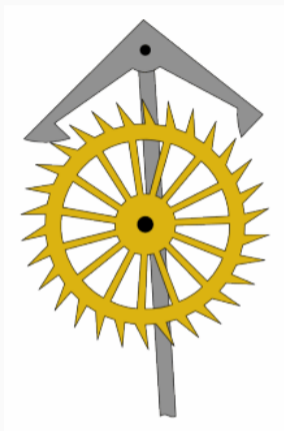
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



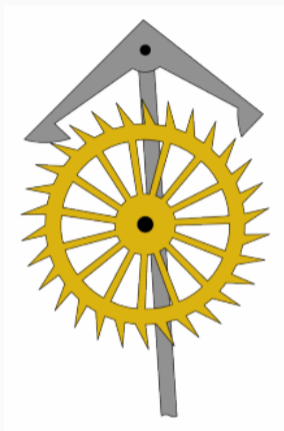
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



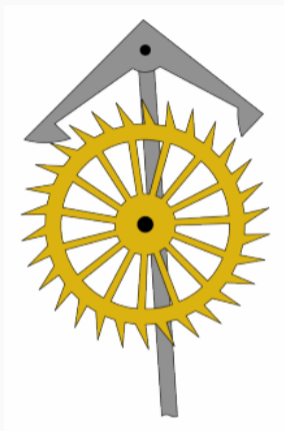
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



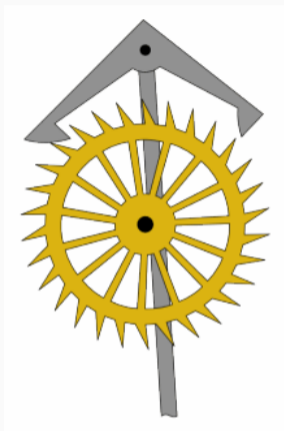
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



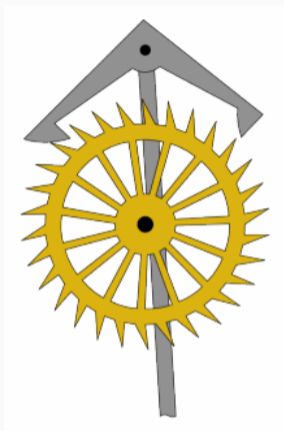
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



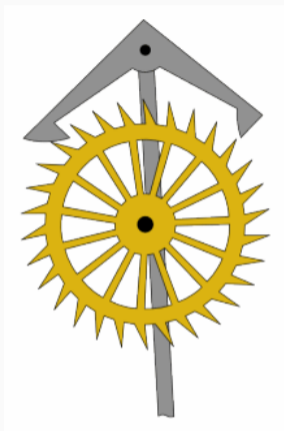
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



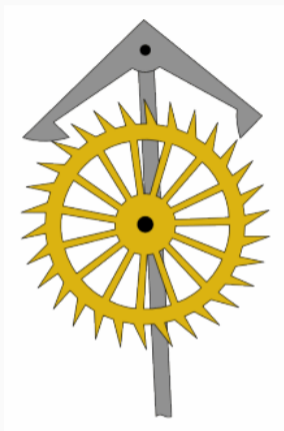
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



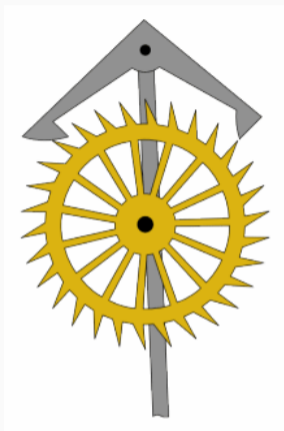
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



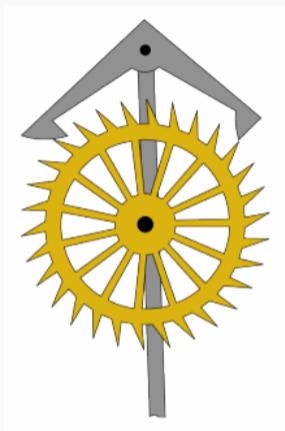
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



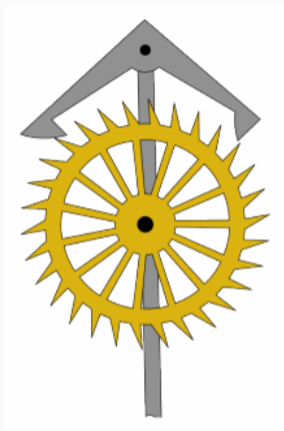
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



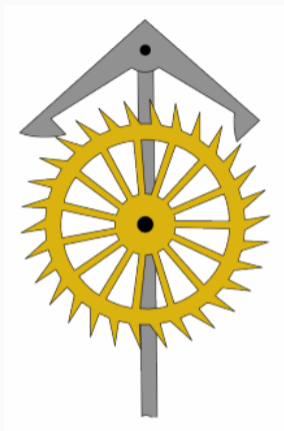
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



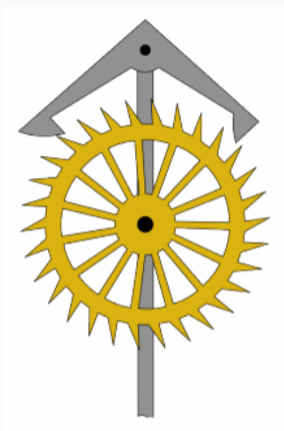
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



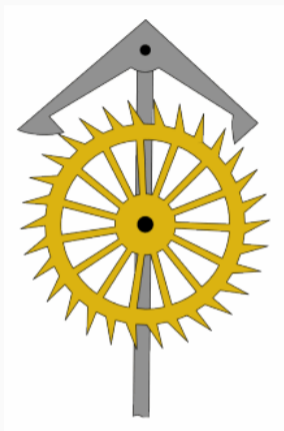
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



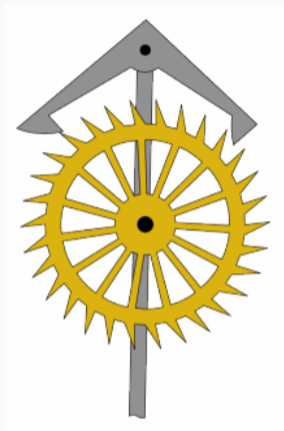
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



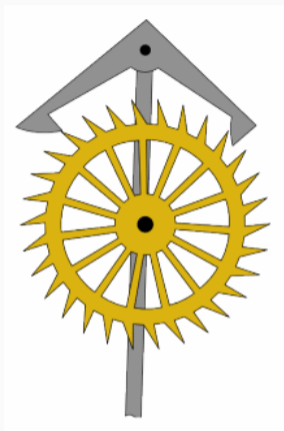
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



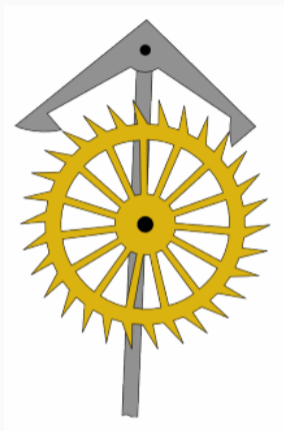
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



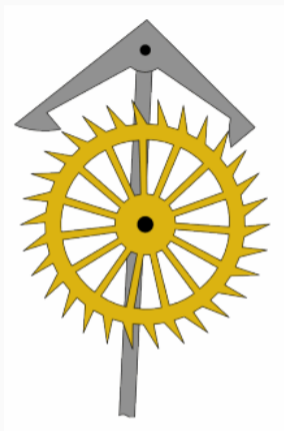
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



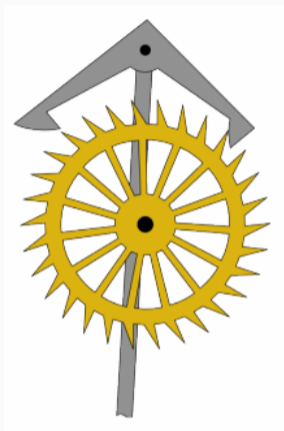
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



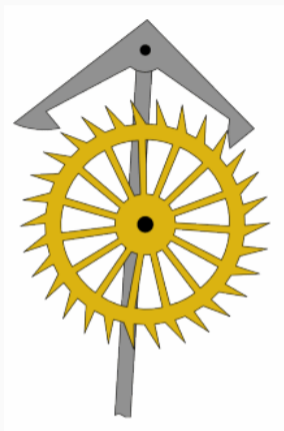
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



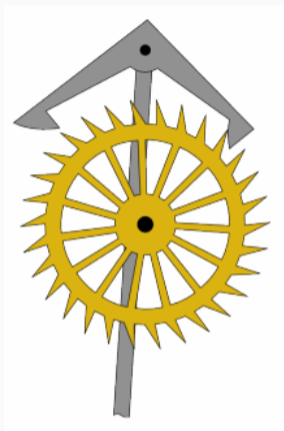
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



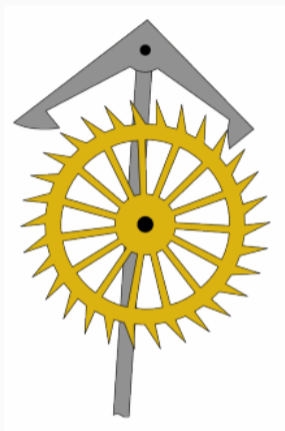
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



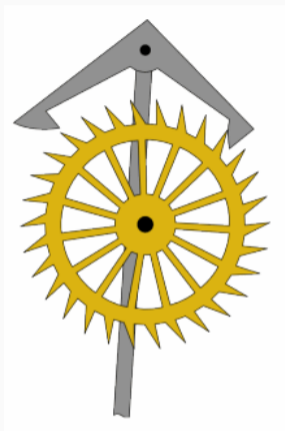
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



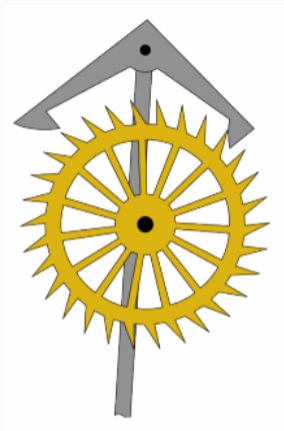
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



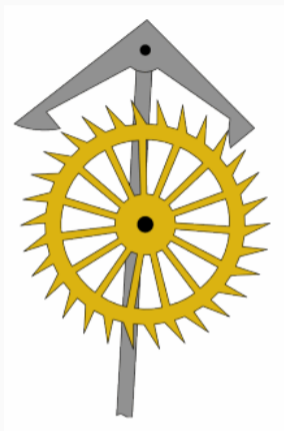
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



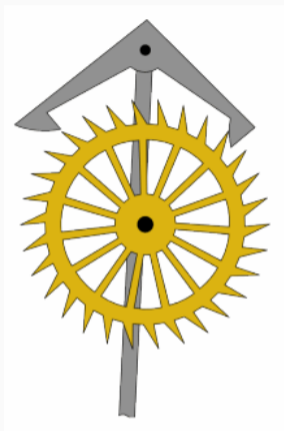
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



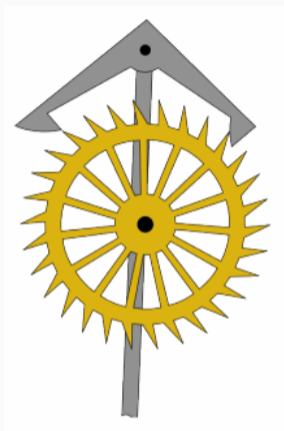
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



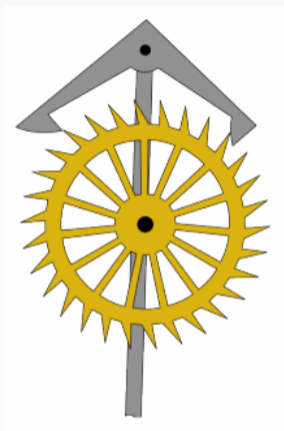
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



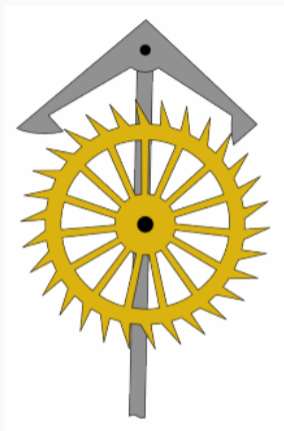
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



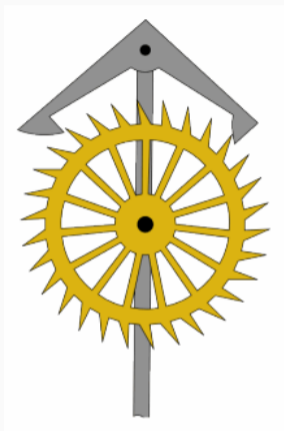
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



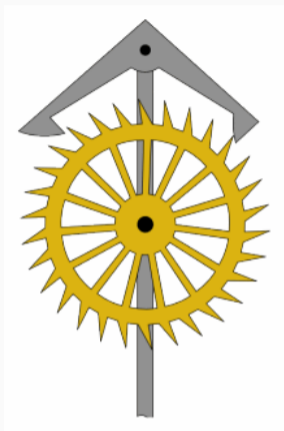
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



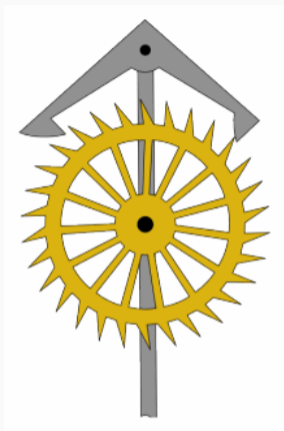
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



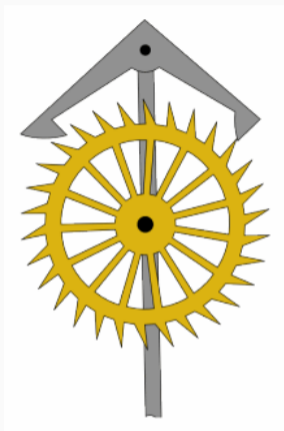
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



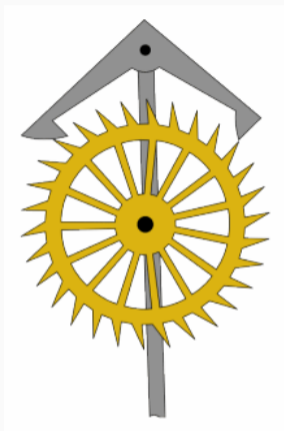
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



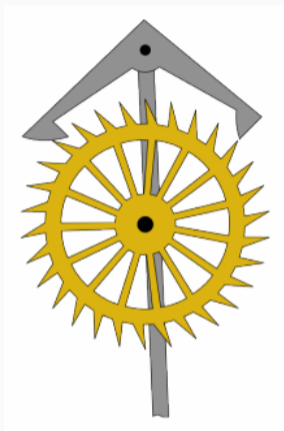
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



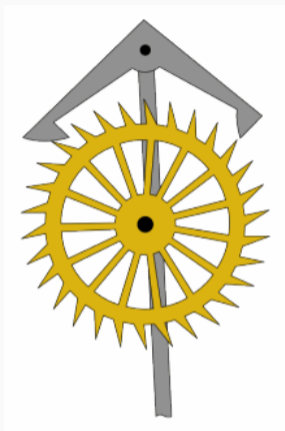
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



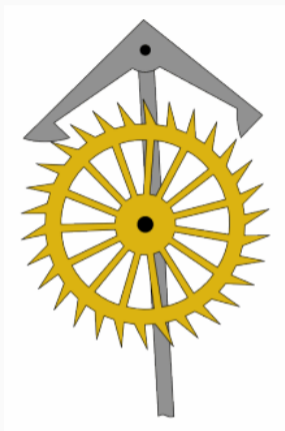
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



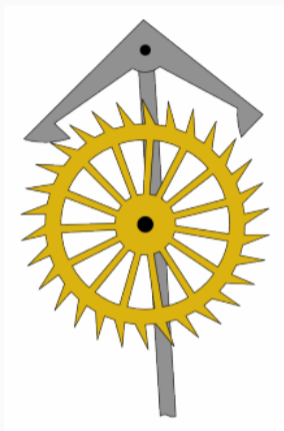
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



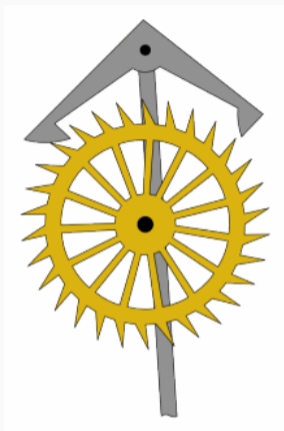
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



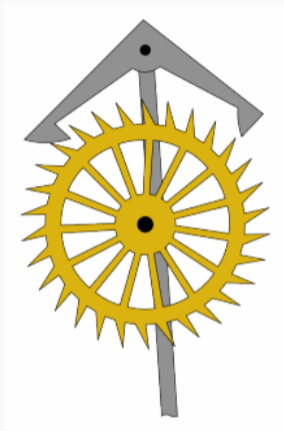
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



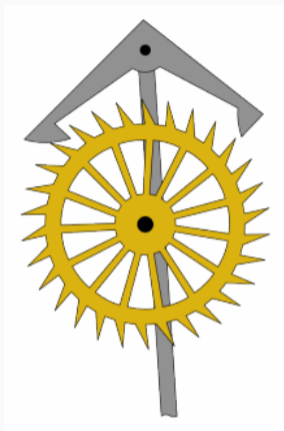
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



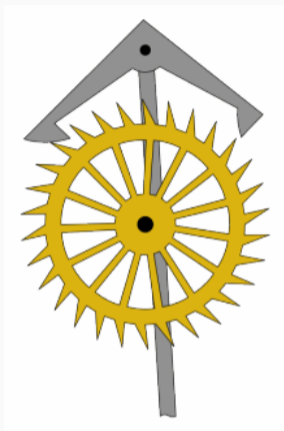
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



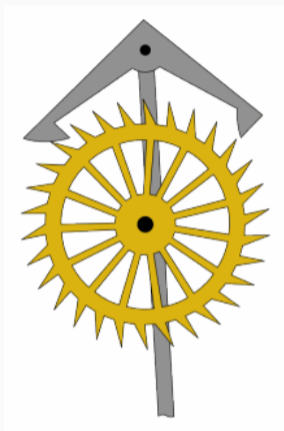
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



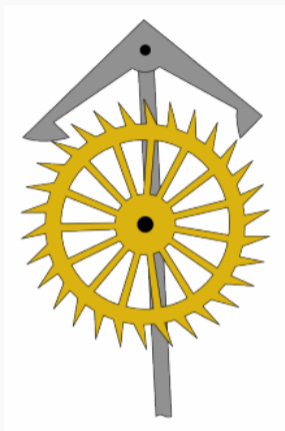
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



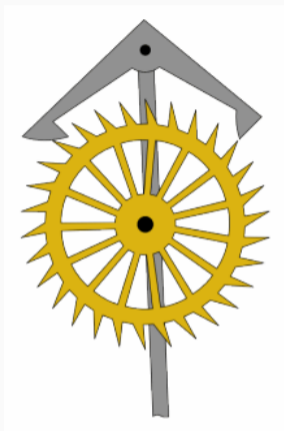
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



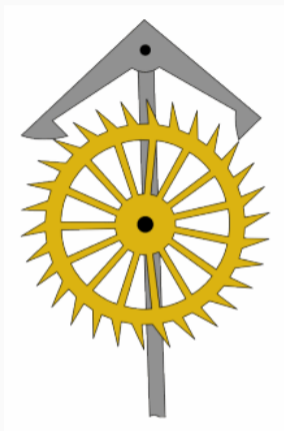
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



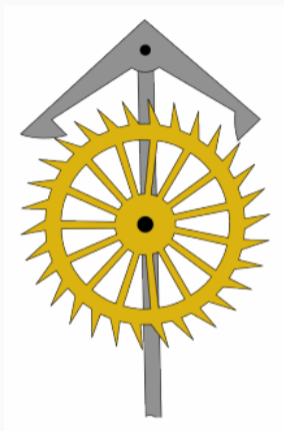
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



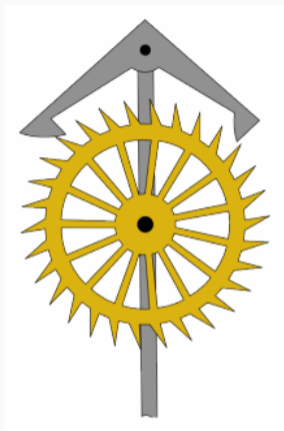
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



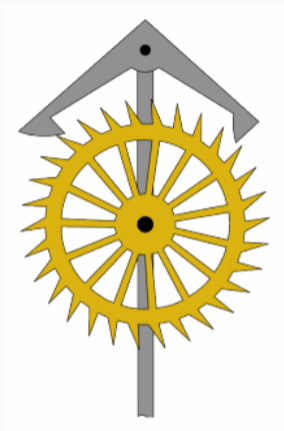
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



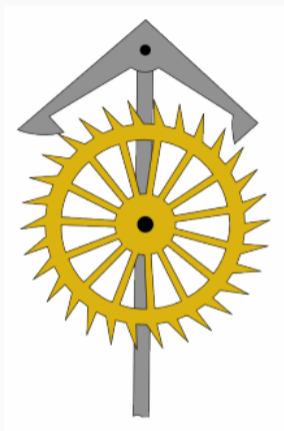
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



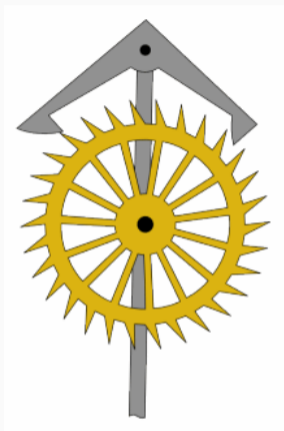
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



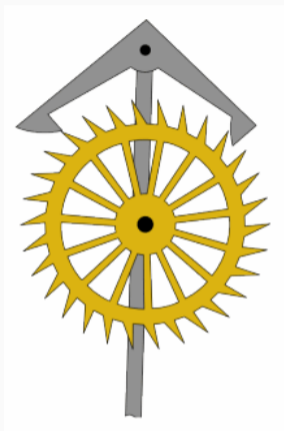
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



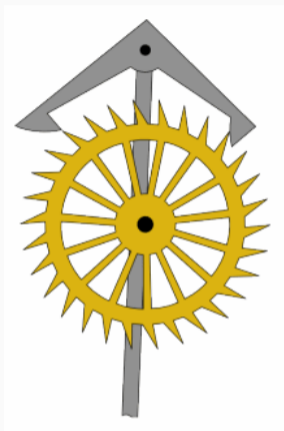
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



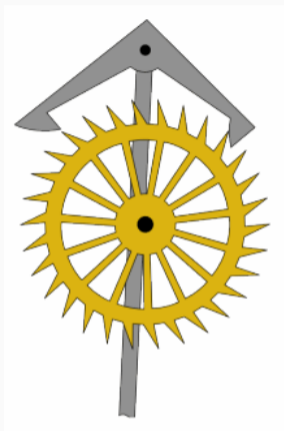
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



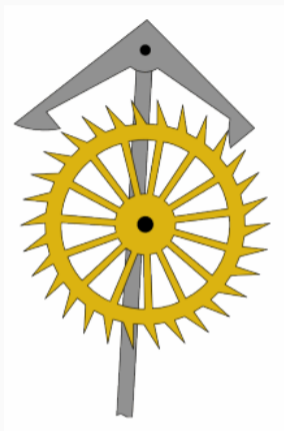
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



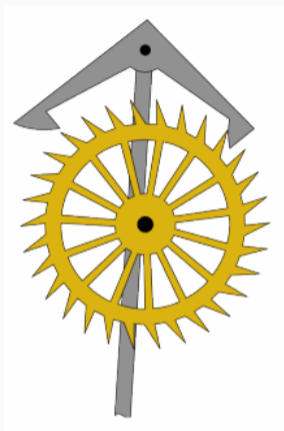
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



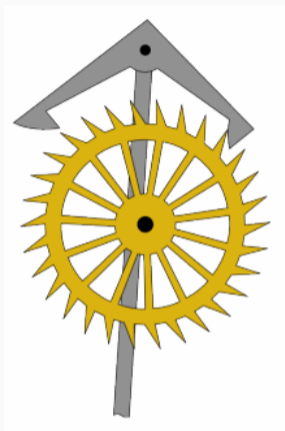
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



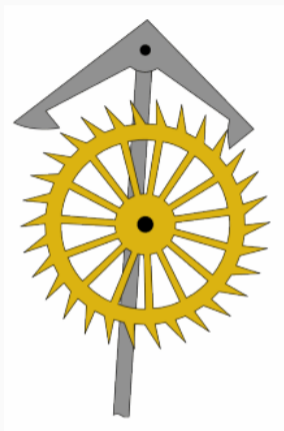
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



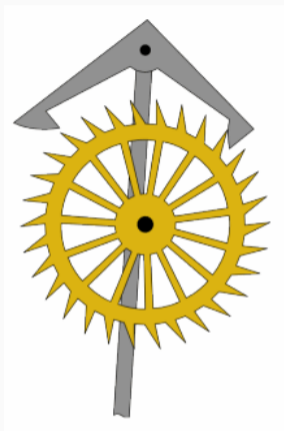
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



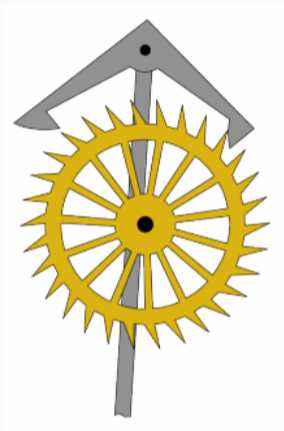
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



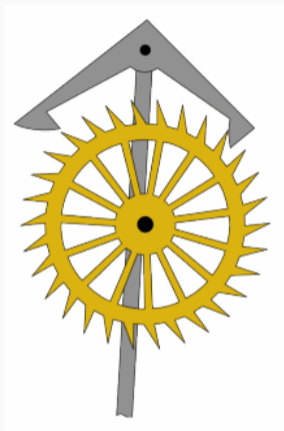
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



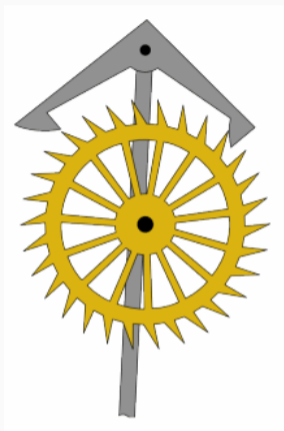
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



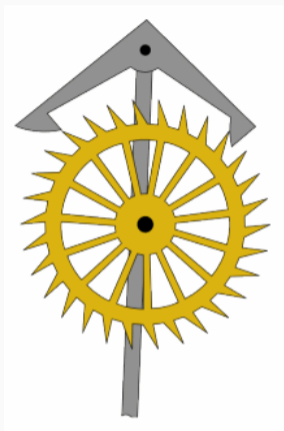
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



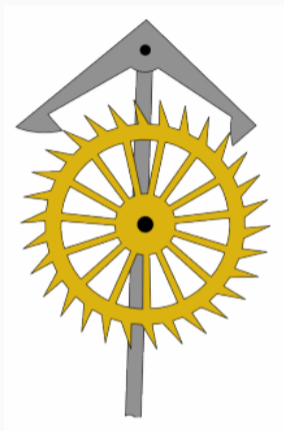
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



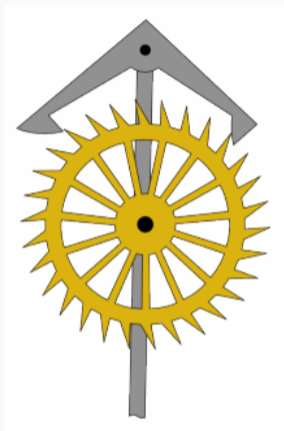
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



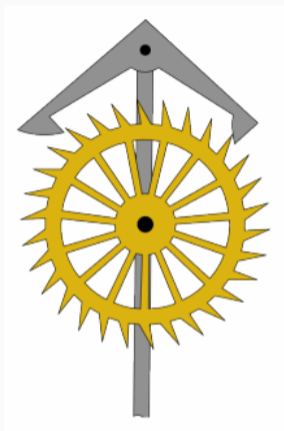
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



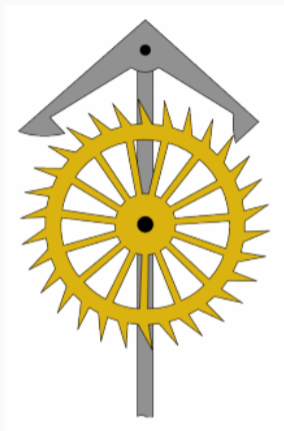
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



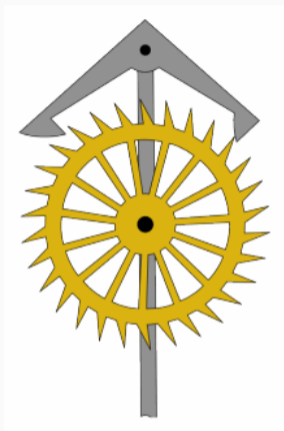
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



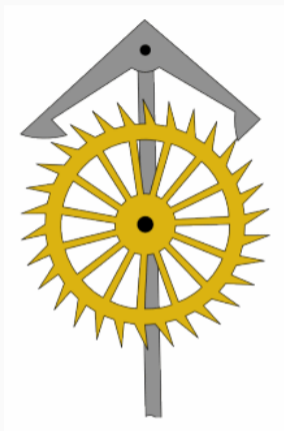
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



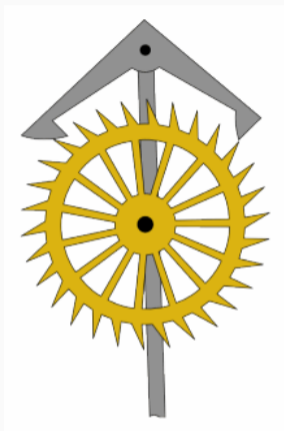
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



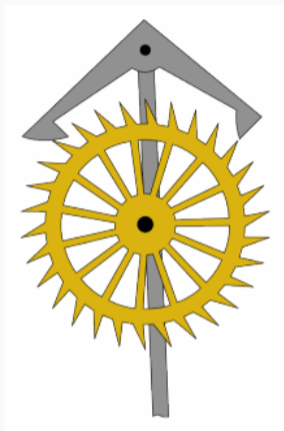
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



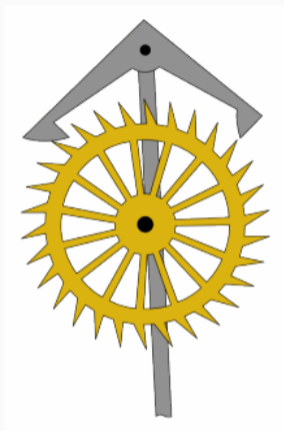
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



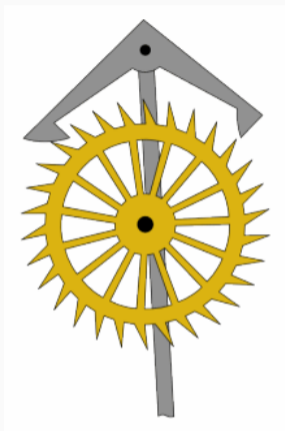
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



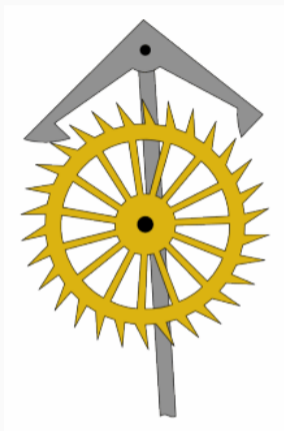
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



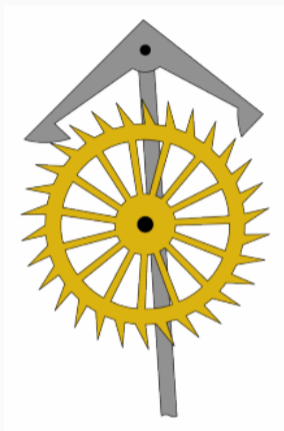
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



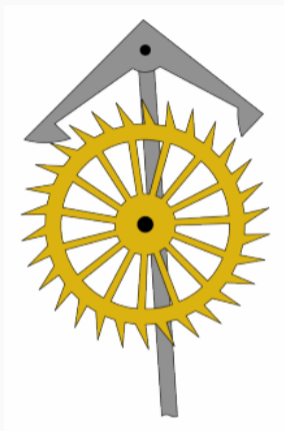
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



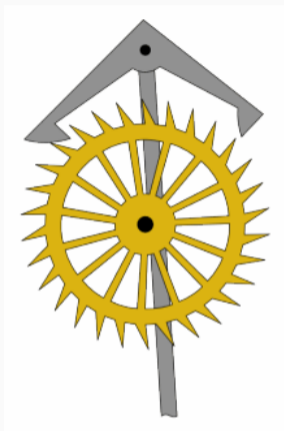
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



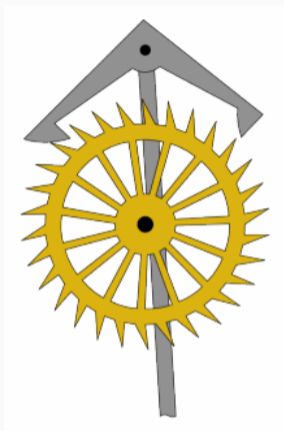
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



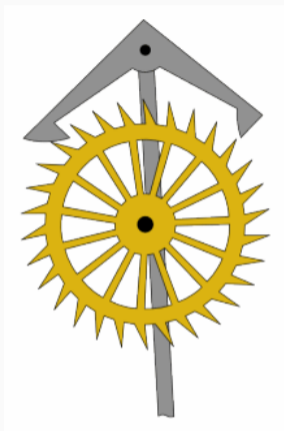
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



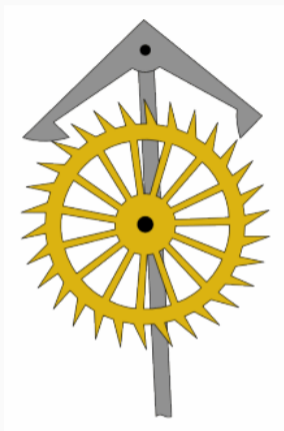
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



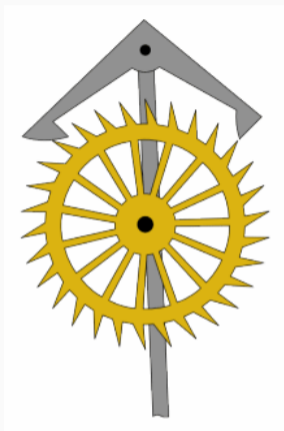
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



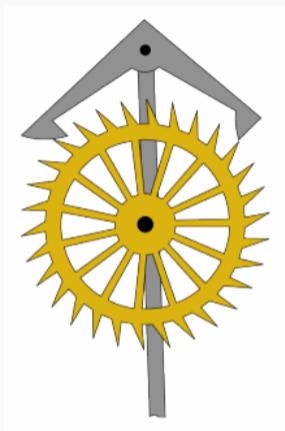
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



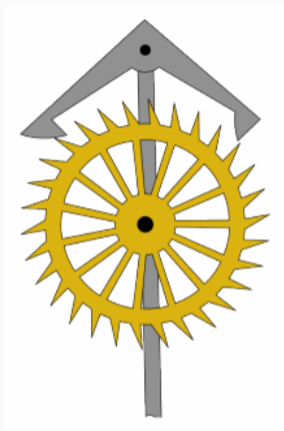
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



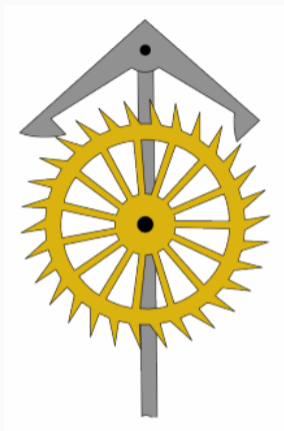
commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif



commons.wikimedia.org/wiki/File:Anchor_escapement_animation_315x478px.gif

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- ▶ Computadores (profissão)

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- ▶ Computadores (profissão)
- ▶ Tabelas de logaritmos

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- ▶ Computadores (profissão)
- ▶ Tabelas de logaritmos
- ▶ Régua de cálculo

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- ▶ Computadores (profissão)
- ▶ Tabelas de logaritmos
- ▶ Régua de cálculo
- ▶ Calculadoras mecânicas

- **Logaritmos (séc XVII)**

- ▶ Mantissa e característica

- » $\log_{10}(120) = \log_{10}(10^2 \times 1.2) = 2 + \log_{10}(1.2)$

- ▶ Computadores (profissão)

- ▶ Tabelas de logaritmos

- ▶ Régua de cálculo

- ▶ Calculadoras mecânicas

- » *O arithmomètre, primeira calculadora mecânica confiável e de sucesso comercial, é do séc XIX*

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 +$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x \approx 2.7856$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2 = .7856$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2 = .7856$$

$$x/100 \approx 10^{.7856}$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2 = .7856$$

$$x/100 \approx 10^{.7856}$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2 = .7856$$

$$x/100 \approx 10^{.7856} \approx 6.1$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

$$\log x/100 \approx 2.7856 - 2 = .7856$$

$$x/100 \approx 10^{.7856} \approx 6.1$$

$$x \approx 610$$

Tabelas de logaritmos (simplificada)

	0	1	2	3	4	5	6	7	8	9
1.	.0000	.0414	.0792	.1139	.1461	.1761	.2041	.2304	.2553	.2788
2.	.3010	.3222	.3424	.3617	.3802	.3979	.4150	.4314	.4472	.4624
3.	.4771	.4914	.5051	.5185	.5315	.5441	.5563	.5682	.5798	.5911
4.	.6021	.6128	.6232	.6335	.6435	.6532	.6628	.6721	.6812	.6902
5.	.6990	.7076	.7160	.7243	.7324	.7404	.7482	.7559	.7634	.7709
6.	.7782	.7853	.7924	.7993	.8062	.8129	.8195	.8261	.8325	.8388
7.	.8451	.8513	.8573	.8633	.8692	.8751	.8808	.8865	.8921	.8976
8.	.9031	.9085	.9138	.9191	.9243	.9294	.9345	.9395	.9445	.9494
9.	.9542	.9590	.9638	.9685	.9731	.9777	.9823	.9868	.9912	.9956

$$x = 13 * 47$$

$$\log x = \log 13 + \log 47$$

$$\log x = 1 + \log 1.3 + 1 + \log 4.7$$

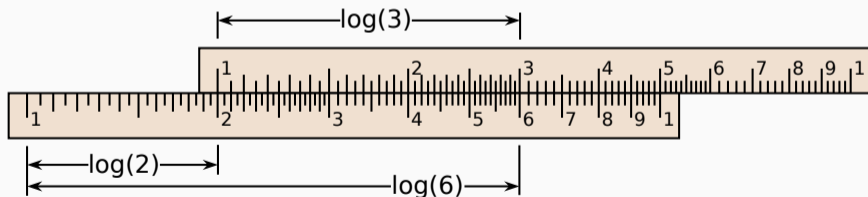
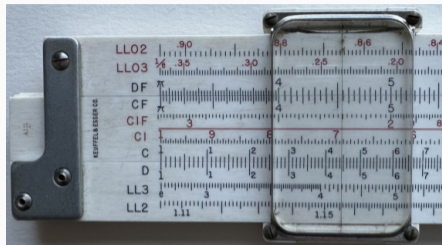
$$\log x \approx 1.1139 + 1.6721 = 2.7856$$

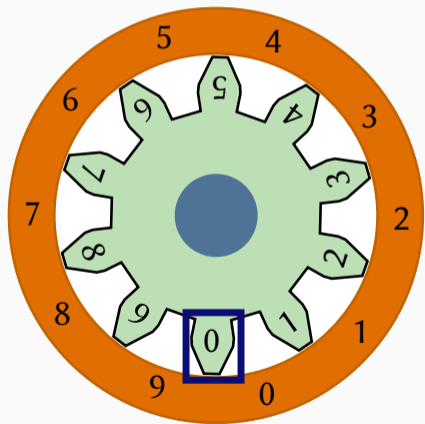
$$\log x/100 \approx 2.7856 - 2 = .7856$$

$$x/100 \approx 10^{.7856} \approx 6.1$$

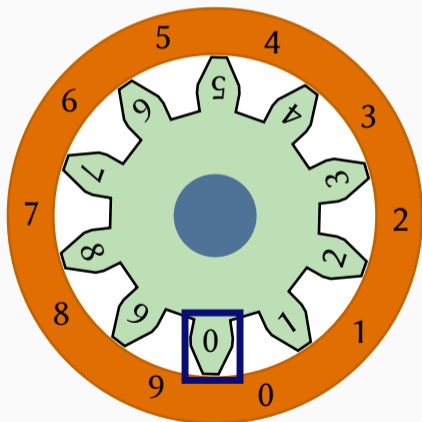
$$x \approx 610 \text{ (o correto é 611)}$$

Réguas de cálculo

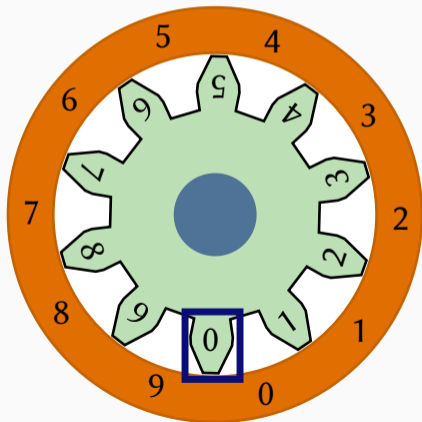


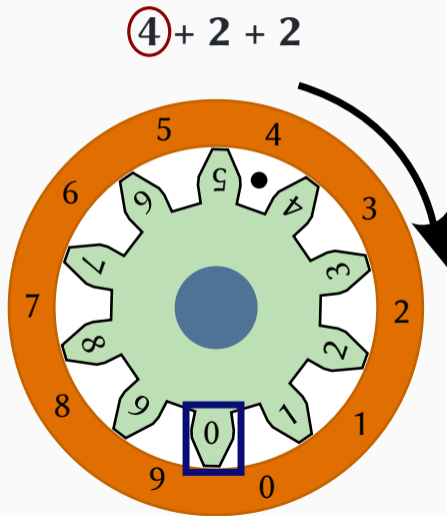


$$4 + 2 + 2$$

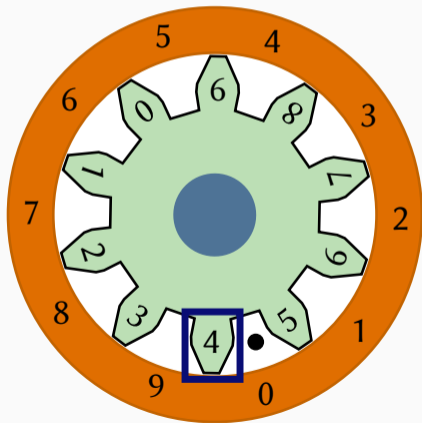


$$\textcircled{4} + 2 + 2$$

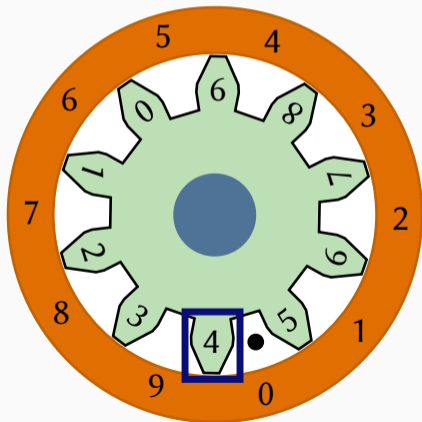




$$\textcircled{4} + 2 + 2$$

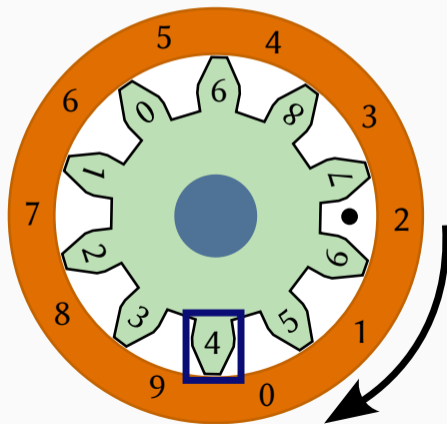


$$4 + \textcircled{2} + 2$$

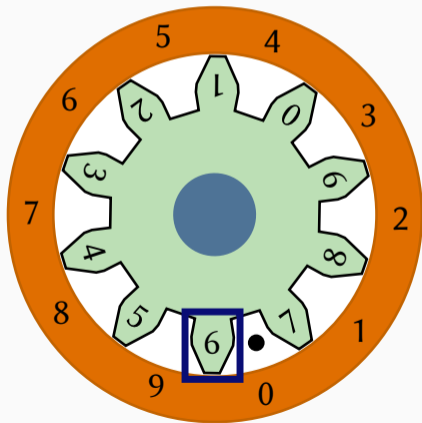


Pascalina

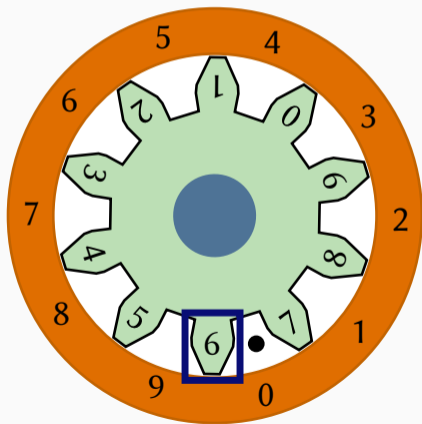
$$4 + \textcircled{2} + 2$$



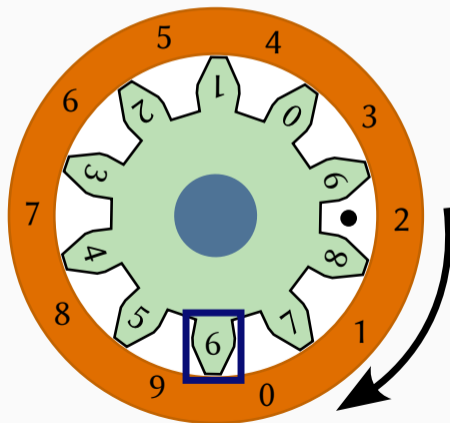
$$4 + \textcircled{2} + 2$$



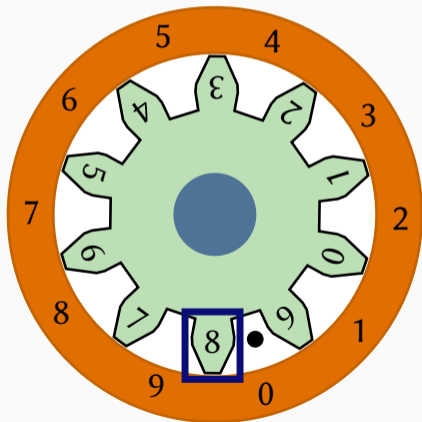
$$4 + 2 + \textcircled{2}$$



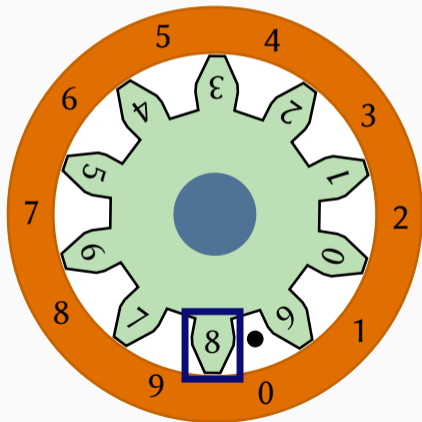
$$4 + 2 + \textcircled{2}$$



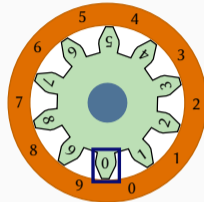
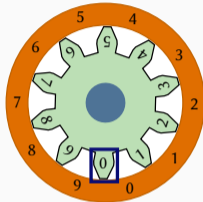
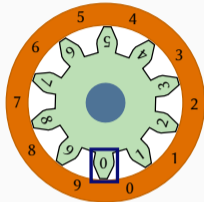
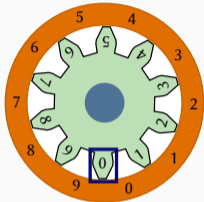
$$4 + 2 + \textcircled{2}$$



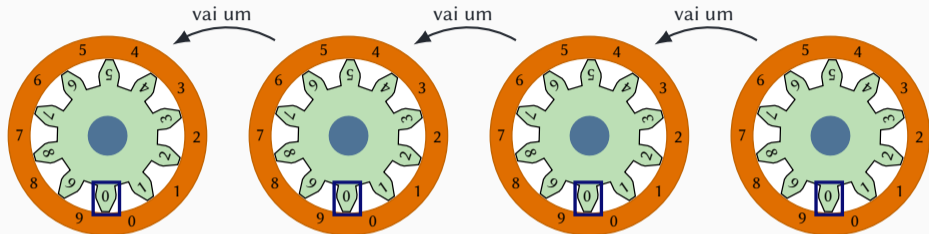
É um acumulador!



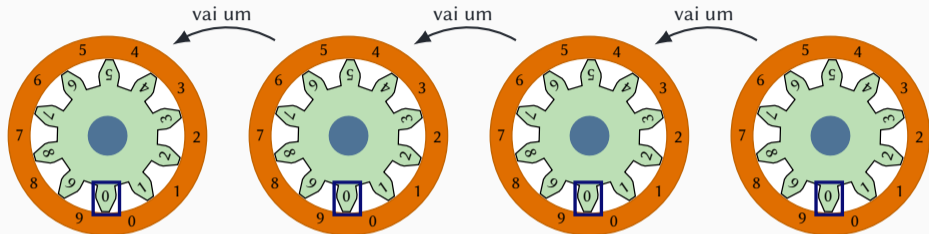
Pascalina



Pascalina



Pascalina



Teoricamente, fazer o “vai um” com engrenagens é simples; na prática, com vários dígitos o esforço mecânico é muito grande. A inovação da Pascalina foi um mecanismo de molas para o “vai um”.

Para mais detalhes:

www.youtube.com/watch?v=3h71HAJWnVU

www.youtube.com/watch?v=nyCrDI7hRpE

Passo

Sistemas analógicos e digitais

- Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto

Sistemas analógicos e digitais

- Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto
 - ▶ Escala **contínua** de valores → sistemas **analógicos**

Sistemas analógicos e digitais

- Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- Em um ábaco não!

Sistemas analógicos e digitais

- **Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto**
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- **Em um ábaco não!**
 - ▶ Durante o movimento de cada peça, o estado do ábaco não faz sentido (a peça não está nem de um lado nem do outro)

Sistemas analógicos e digitais

- **Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto**
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- **Em um ábaco não!**
 - ▶ Durante o movimento de cada peça, o estado do ábaco não faz sentido (a peça não está nem de um lado nem do outro)
- **Ao fazer uma soma no papel também não!**

Sistemas analógicos e digitais

- **Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto**
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- **Em um ábaco não!**
 - ▶ Durante o movimento de cada peça, o estado do ábaco não faz sentido (a peça não está nem de um lado nem do outro)
- **Ao fazer uma soma no papel também não!**
 - ▶ Enquanto os dígitos estão sendo desenhados, o estado da expressão no papel não faz sentido

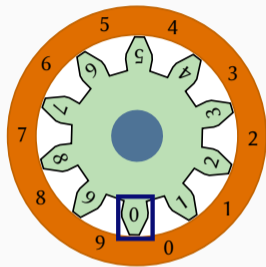
Sistemas analógicos e digitais

- **Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto**
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- **Em um ábaco não!**
 - ▶ Durante o movimento de cada peça, o estado do ábaco não faz sentido (a peça não está nem de um lado nem do outro)
- **Ao fazer uma soma no papel também não!**
 - ▶ Enquanto os dígitos estão sendo desenhados, o estado da expressão no papel não faz sentido
- **O processamento é feito em **passos** descontínuos**

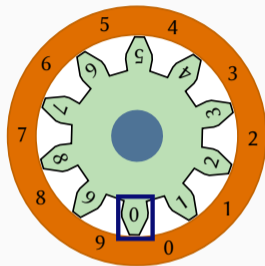
Sistemas analógicos e digitais

- **Em uma balança, relógio ou régua de cálculo, o movimento das partes vai indicando valores cada vez mais próximos do resultado correto**
 - ▶ Escala **contínua** de valores → sistemas **analógicos**
- **Em um ábaco não!**
 - ▶ Durante o movimento de cada peça, o estado do ábaco não faz sentido (a peça não está nem de um lado nem do outro)
- **Ao fazer uma soma no papel também não!**
 - ▶ Enquanto os dígitos estão sendo desenhados, o estado da expressão no papel não faz sentido
- **O processamento é feito em **passos** descontínuos**
 - ▶ sistemas **digitais**

Pascalina passo a passo

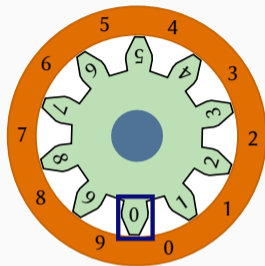


Pascalina passo a passo



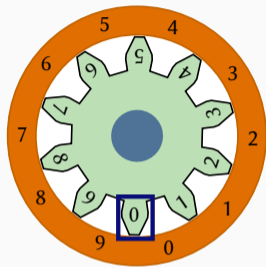
- Cada vez que movemos a engrenagem, realizamos um **passo**

Pascalina passo a passo



- Cada vez que movemos a engrenagem, realizamos um **passo**
 - ▶ Enquanto a engrenagem está em movimento (ou seja, durante a execução do passo), o estado da pascalina não faz sentido

Pascalina passo a passo



- Cada vez que movemos a engrenagem, realizamos um **passo**
 - ▶ Enquanto a engrenagem está em movimento (ou seja, durante a execução do passo), o estado da pascalina não faz sentido
- Cada passo → uma operação de um dígito

- O mecanismo de Anticítera era um computador **analógico**

Sistemas analógicos e digitais

- O mecanismo de Anticítera era um computador **analógico**
- A régua de cálculo é um instrumento **analógico**

Sistemas analógicos e digitais

- O mecanismo de Anticítera era um computador **analógico**
- A régua de cálculo é um instrumento **analógico**
- O ábaco é um instrumento **digital**

Sistemas analógicos e digitais

- O mecanismo de Anticítera era um computador **analógico**
- A régua de cálculo é um instrumento **analógico**
- O ábaco é um instrumento **digital**
- As calculadoras mecânicas são sistemas **digitais**

Sistemas analógicos e digitais

- O mecanismo de Anticítera era um computador **analógico**
- A régua de cálculo é um instrumento **analógico**
- O ábaco é um instrumento **digital**
- As calculadoras mecânicas são sistemas **digitais**

Sistemas digitais têm a noção de **passo**

- A pascalina processa **um dígito** a cada passo

Arithmomètre

- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo

Arithmomètre

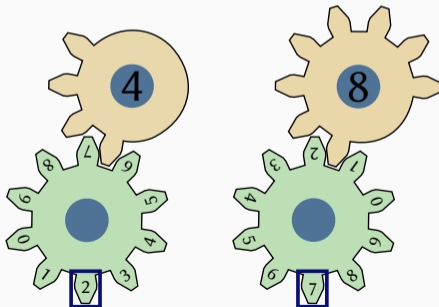
- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela

Arithmomètre

- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela
 - » *Evidentemente, muito mais rápido*

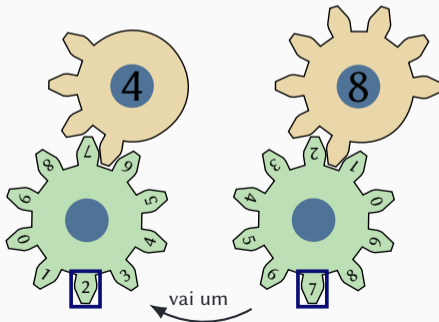
Arithmomètre

- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela
 - » *Evidentemente, muito mais rápido*



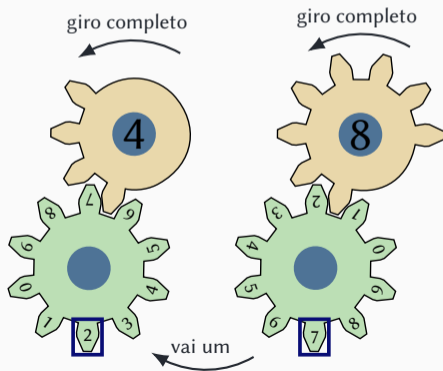
Arithmomètre

- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela
 - » *Evidentemente, muito mais rápido*



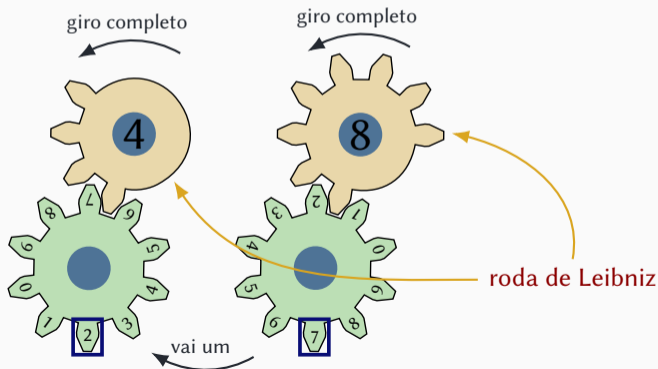
Arithmomètre

- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela
 - » *Evidentemente, muito mais rápido*

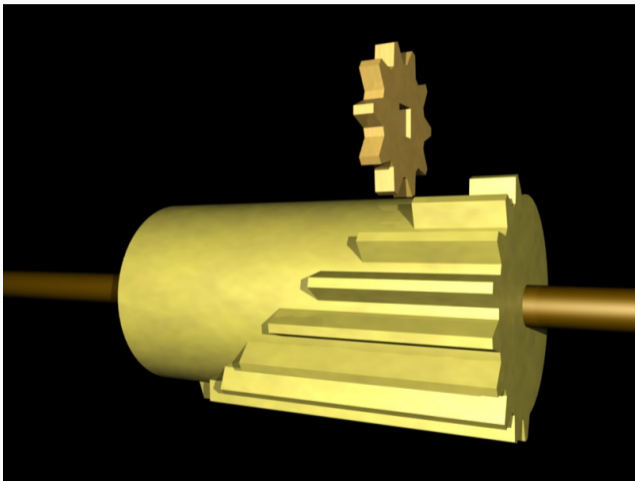


Arithmomètre

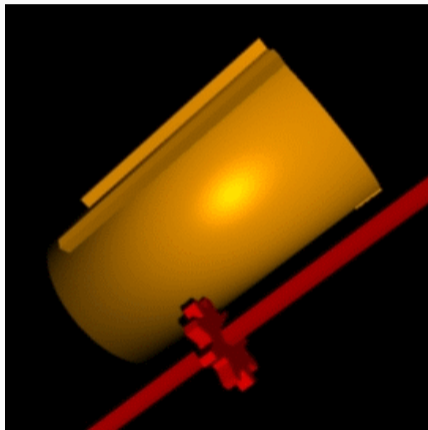
- A pascalina processa **um dígito** a cada passo
- O arithmomètre processa **todos os dígitos** a cada passo
 - ▶ Um passo → um giro completo de uma manivela
 - » *Evidentemente, muito mais rápido*



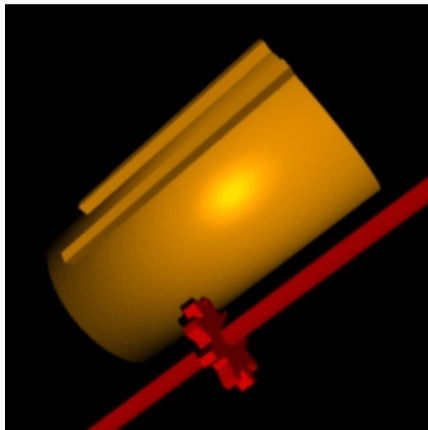
Roda de Leibniz



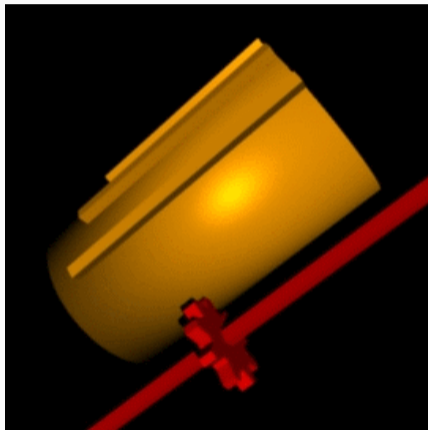
Roda de Leibniz



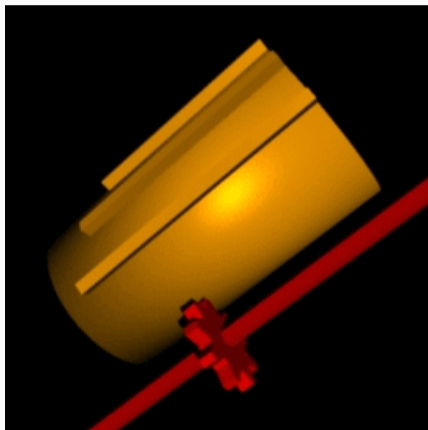
Roda de Leibniz



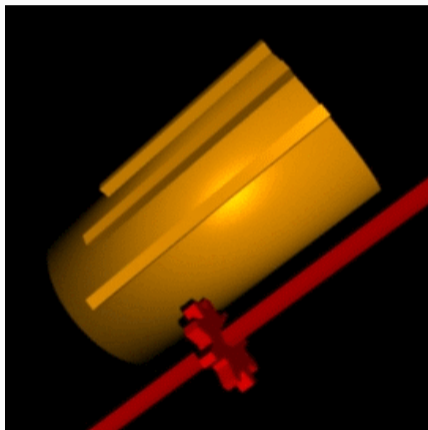
Roda de Leibniz



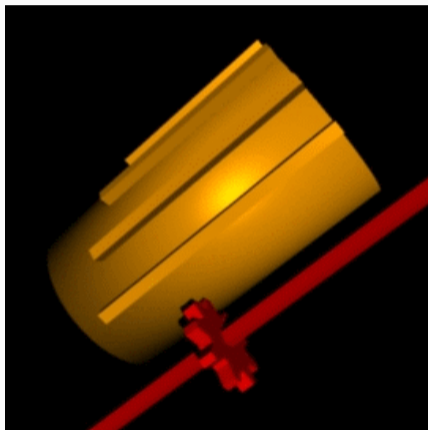
Roda de Leibniz



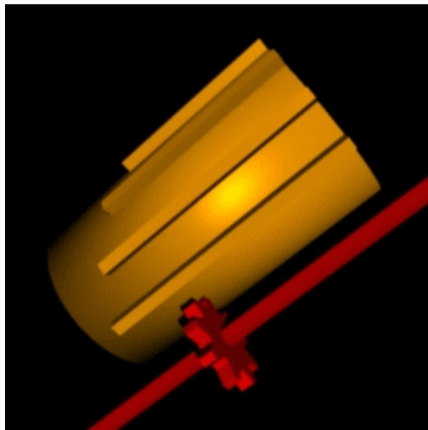
Roda de Leibniz



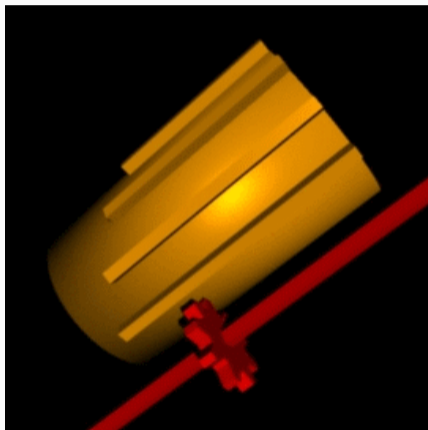
Roda de Leibniz



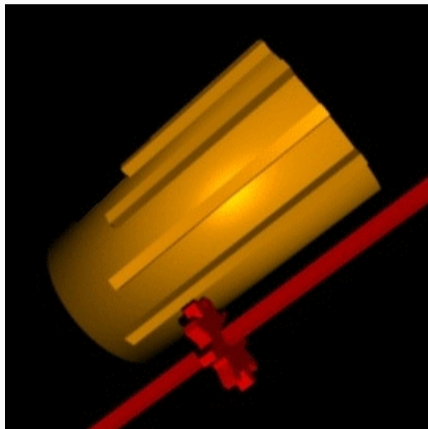
Roda de Leibniz



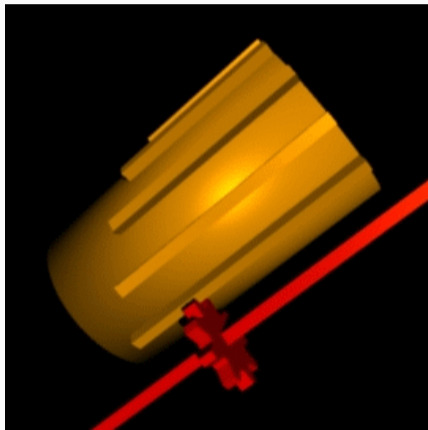
Roda de Leibniz



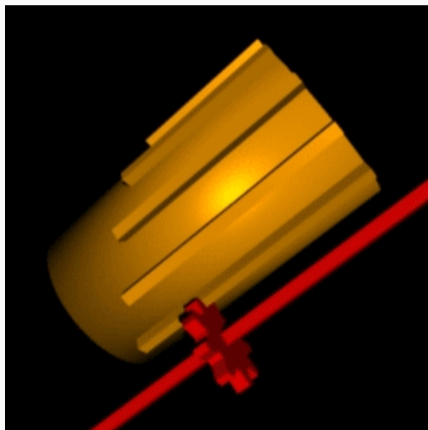
Roda de Leibniz



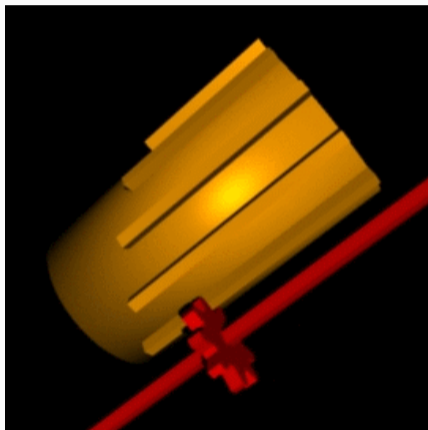
Roda de Leibniz



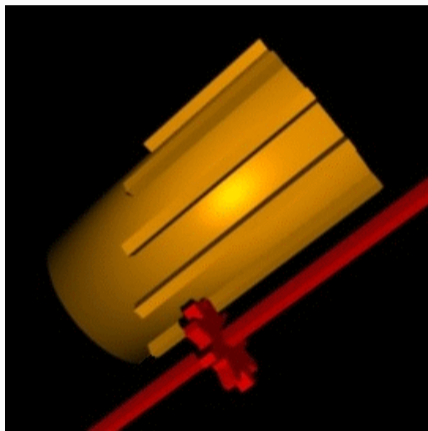
Roda de Leibniz



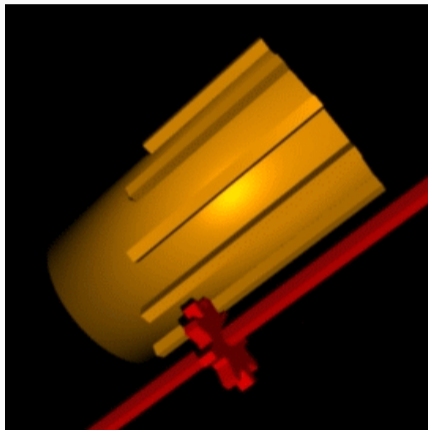
Roda de Leibniz



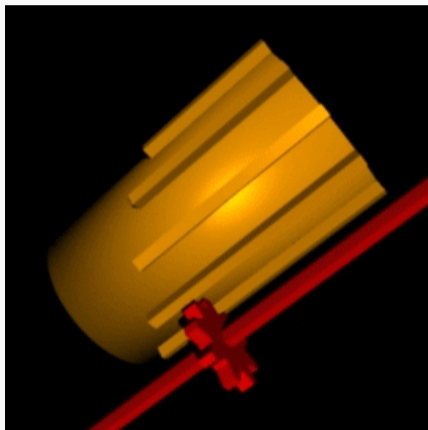
Roda de Leibniz



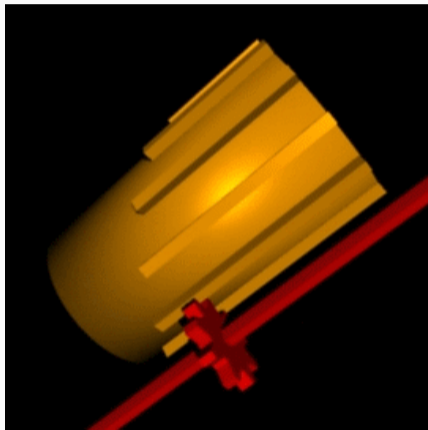
Roda de Leibniz



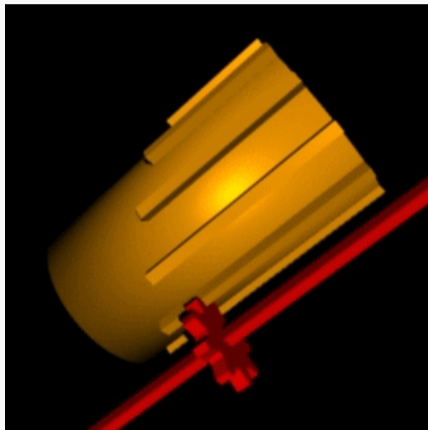
Roda de Leibniz



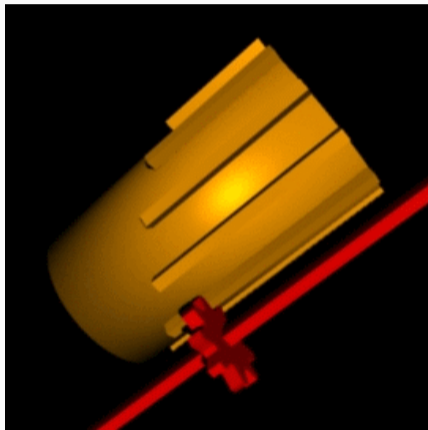
Roda de Leibniz



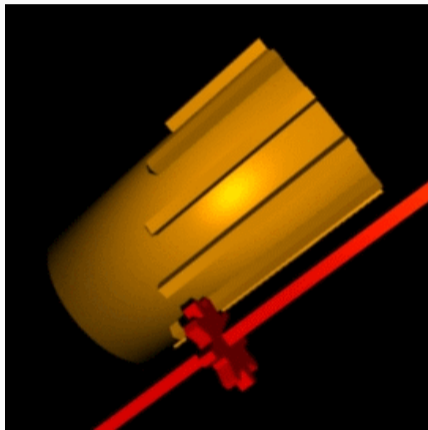
Roda de Leibniz



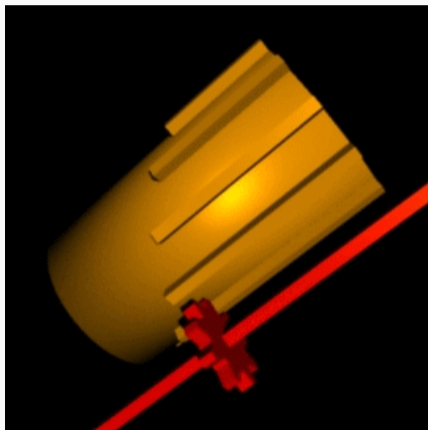
Roda de Leibniz



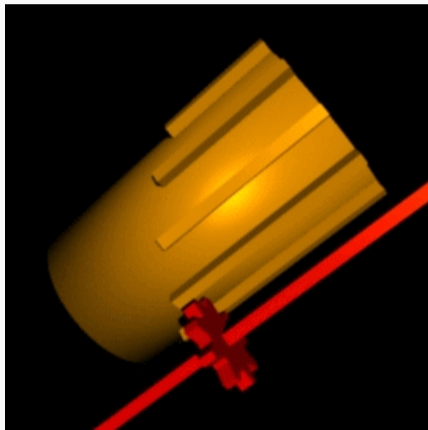
Roda de Leibniz



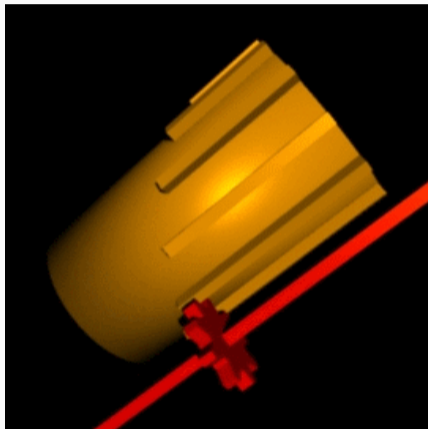
Roda de Leibniz



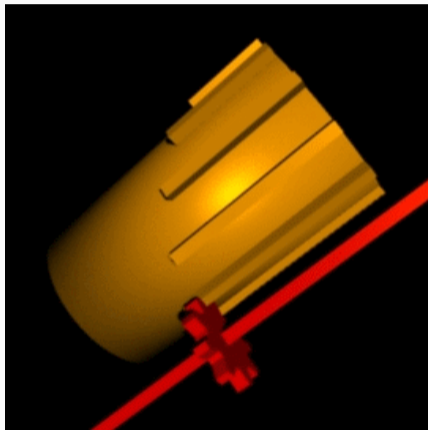
Roda de Leibniz



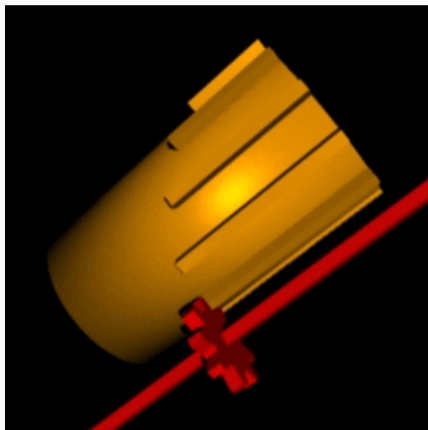
Roda de Leibniz



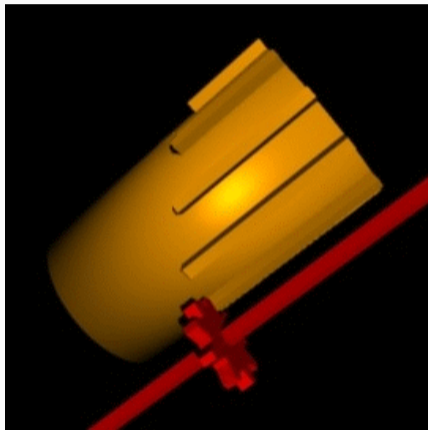
Roda de Leibniz



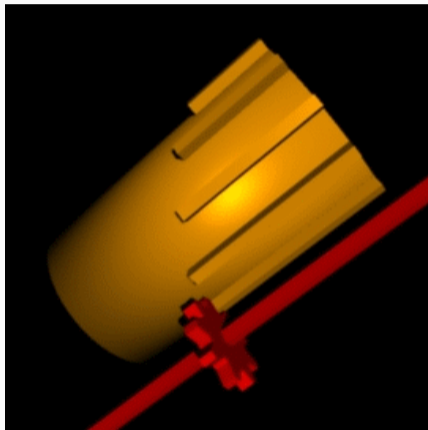
Roda de Leibniz



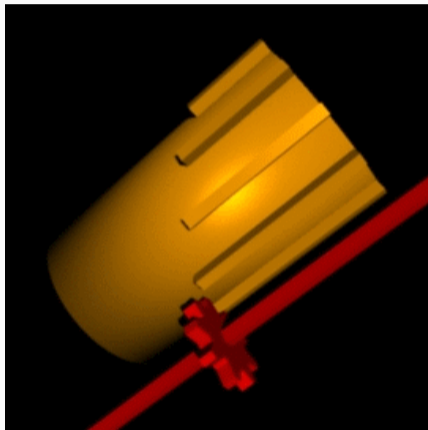
Roda de Leibniz



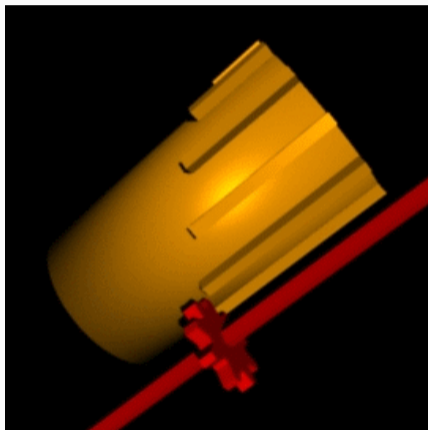
Roda de Leibniz



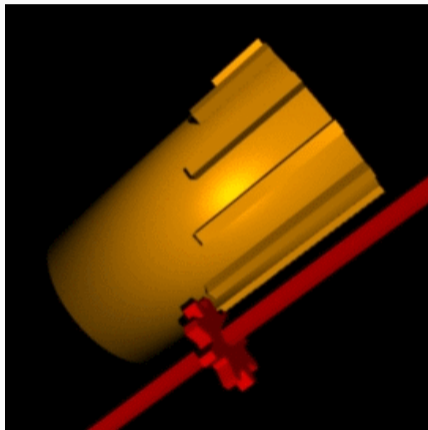
Roda de Leibniz



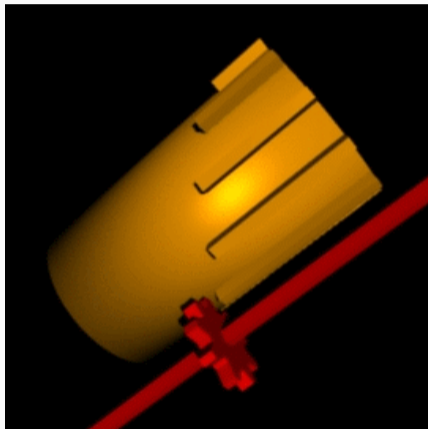
Roda de Leibniz



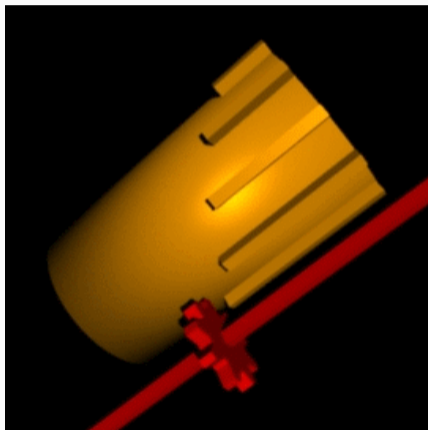
Roda de Leibniz



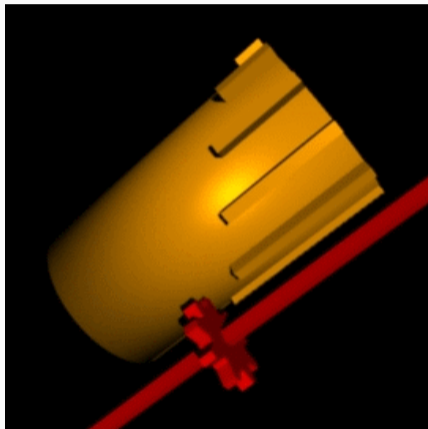
Roda de Leibniz



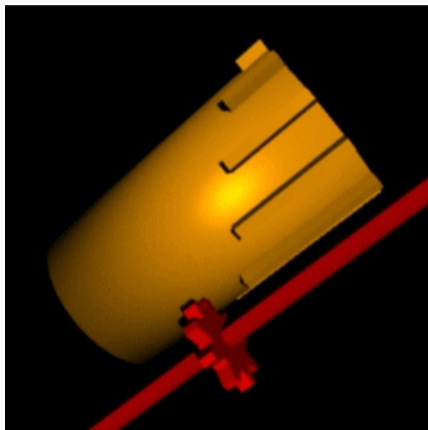
Roda de Leibniz



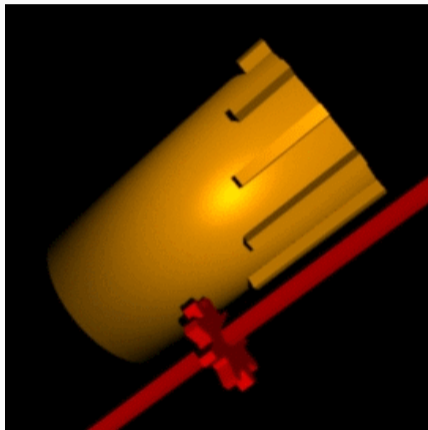
Roda de Leibniz



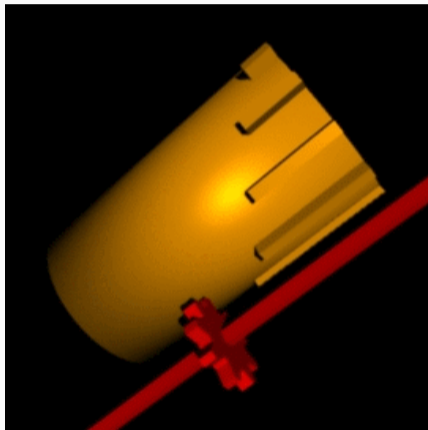
Roda de Leibniz



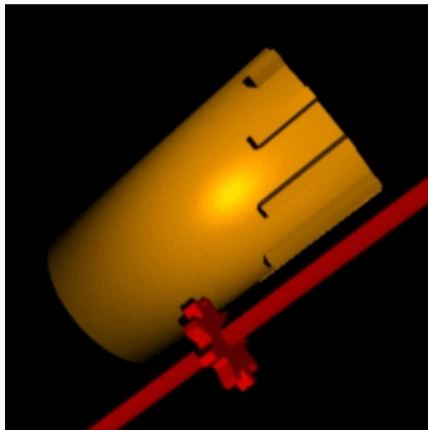
Roda de Leibniz



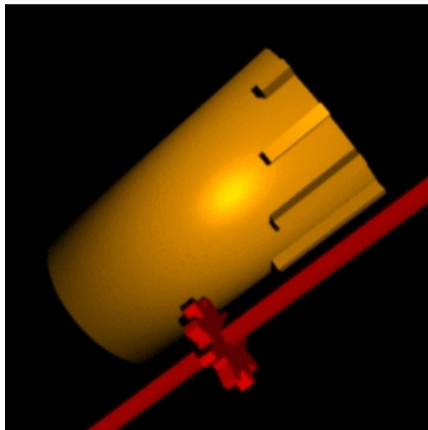
Roda de Leibniz



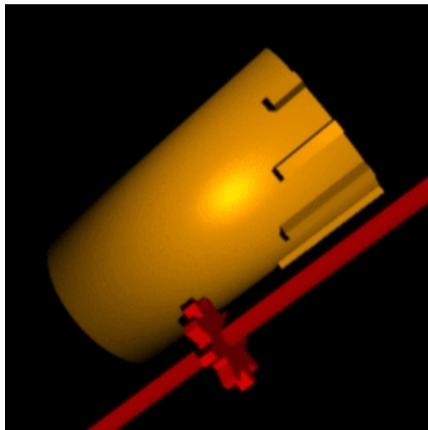
Roda de Leibniz



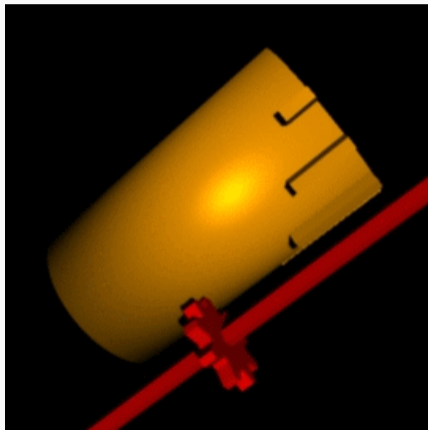
Roda de Leibniz



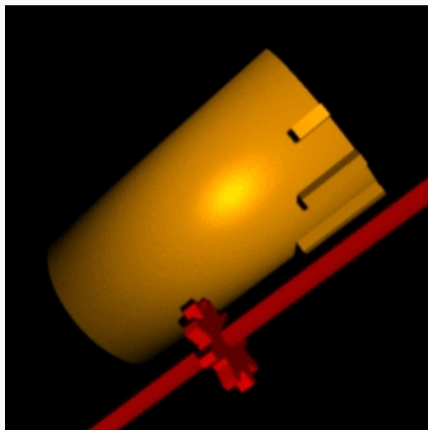
Roda de Leibniz



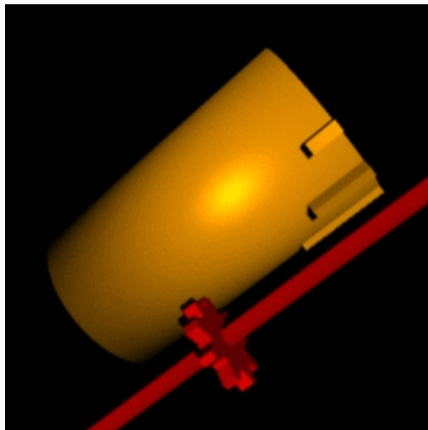
Roda de Leibniz



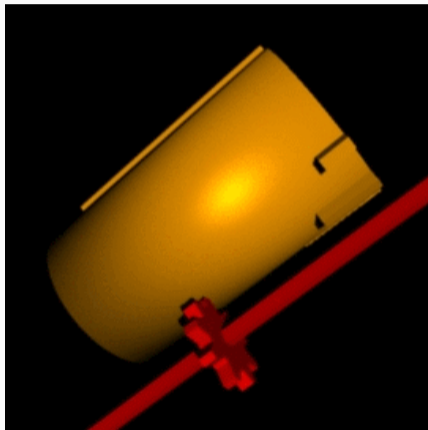
Roda de Leibniz



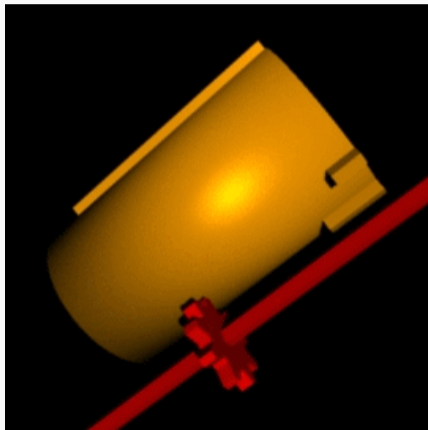
Roda de Leibniz



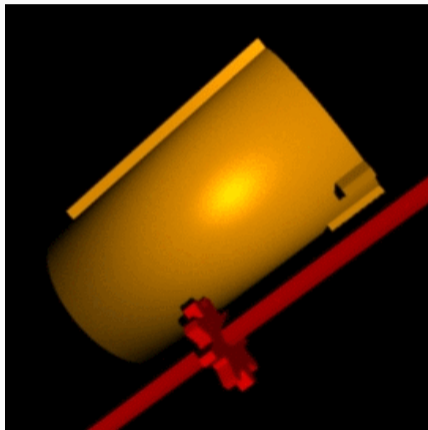
Roda de Leibniz



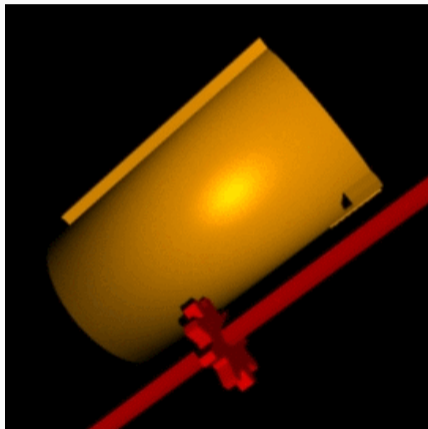
Roda de Leibniz



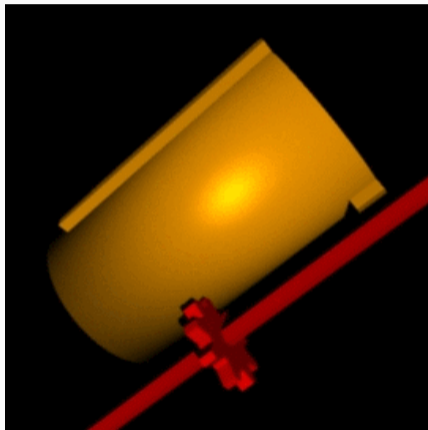
Roda de Leibniz



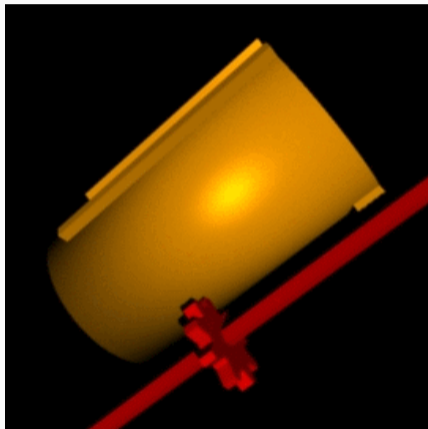
Roda de Leibniz



Roda de Leibniz



Roda de Leibniz





commons.wikimedia.org/wiki/File:Vue_Detaillée_Platines_de_l'Arithmomètre.jpg

Veja também:

www.youtube.com/watch?v=nyCrDI7hRpE

www.youtube.com/watch?v=YXMUjco8onQ

Tear de Jacquard

Tear de Jacquard

WTF “tear”?

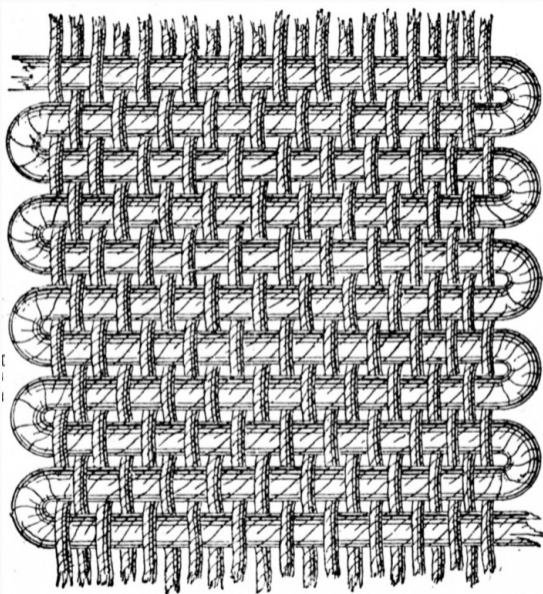
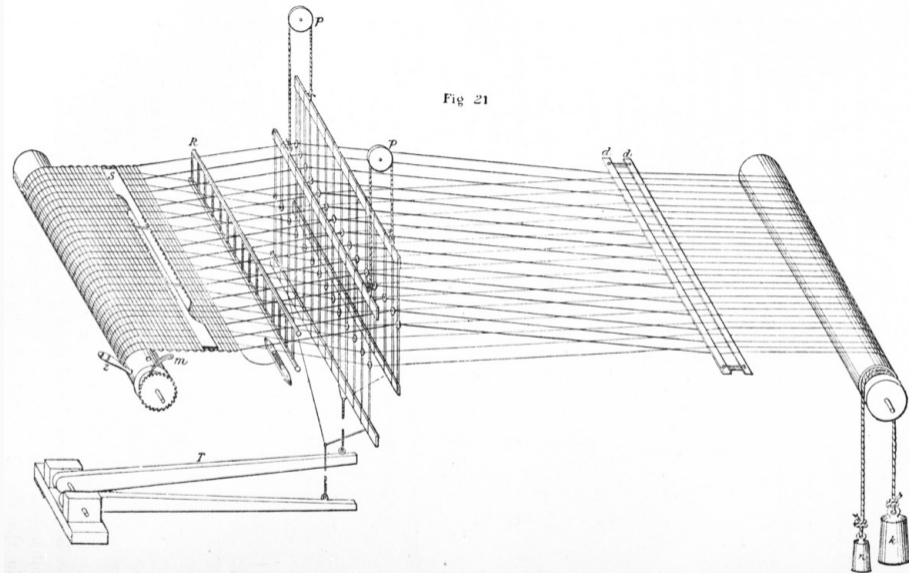
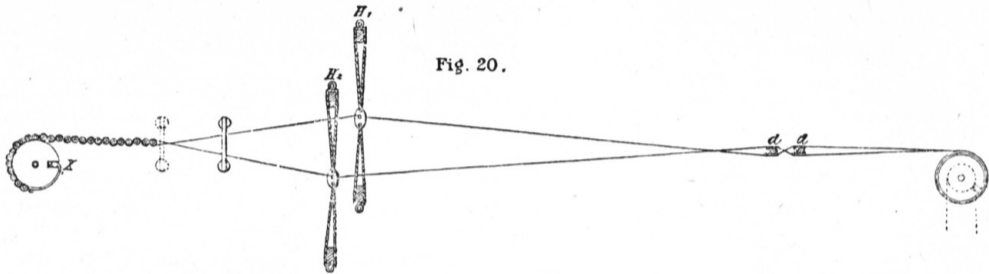
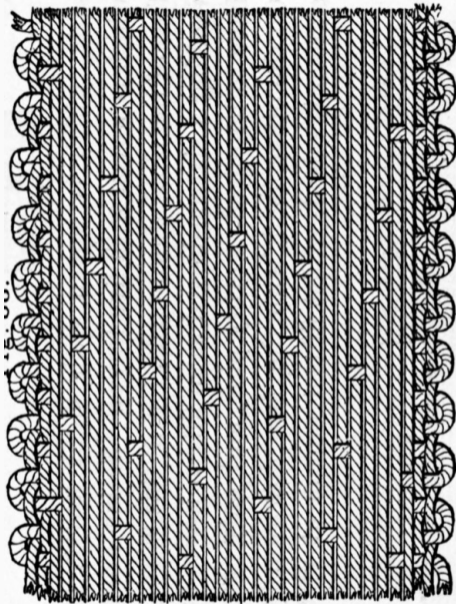


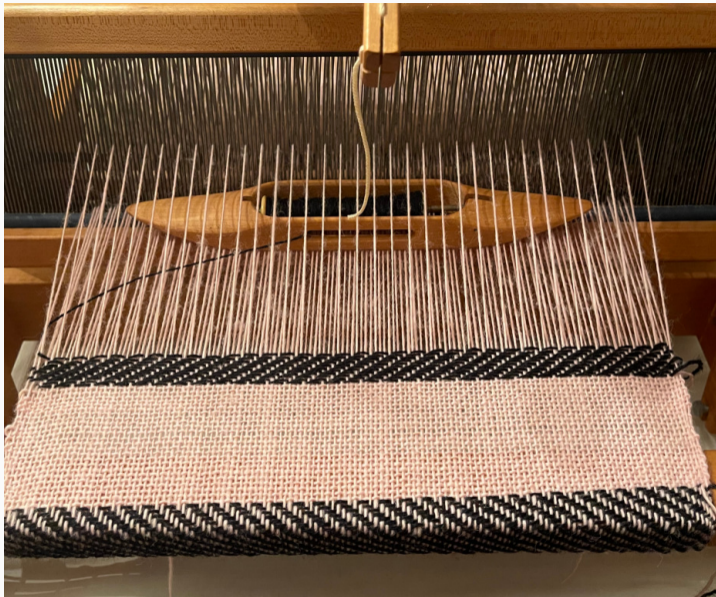


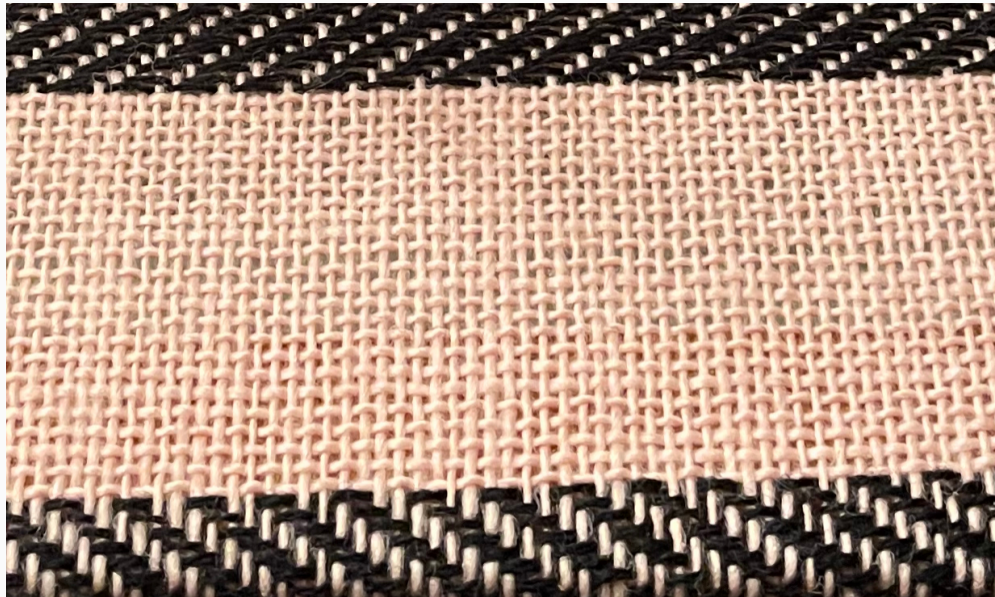
Fig 21

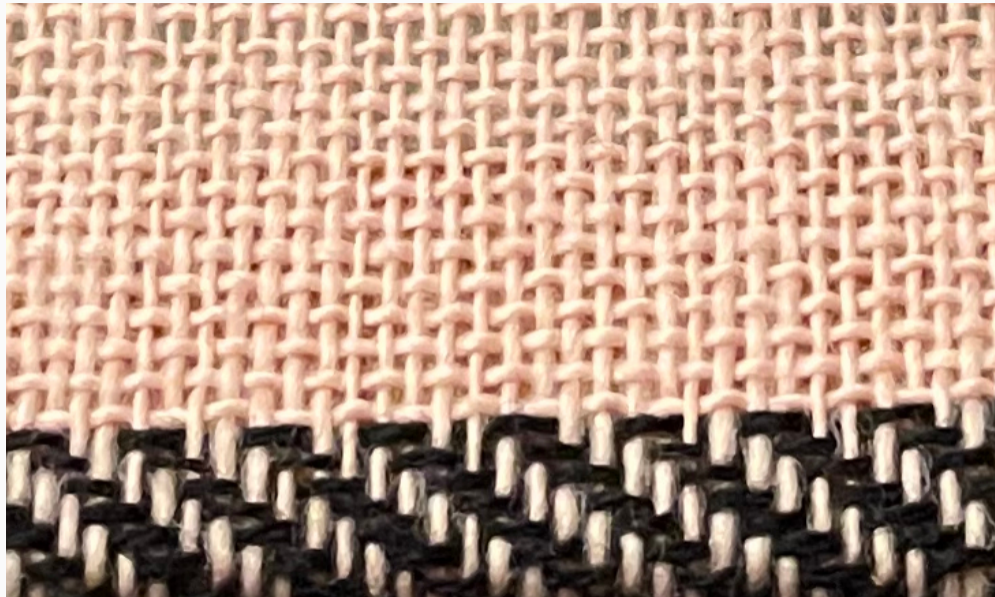


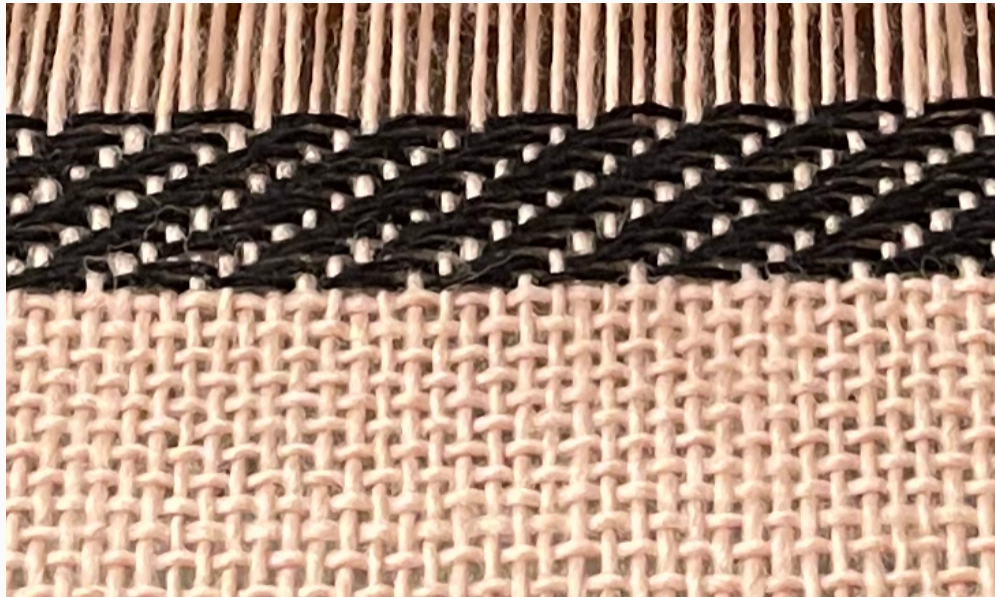












Tear de Jacquard

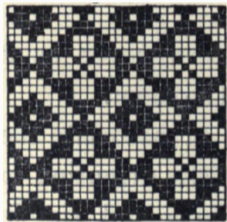
- **Tramas complexas envolvem mais conjuntos de linhas ao invés de apenas dois**

Tear de Jacquard

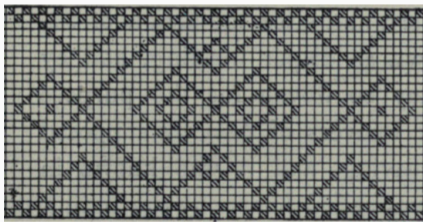
- **Tramas complexas envolvem mais conjuntos de linhas ao invés de apenas dois**
- **Com um número não muito grande de conjuntos, é possível gerar vários padrões geométricos (repetitivos) diferentes**

Tear de Jacquard

- Tramas complexas envolvem mais conjuntos de linhas ao invés de apenas dois
- Com um número não muito grande de conjuntos, é possível gerar vários padrões geométricos (repetitivos) diferentes



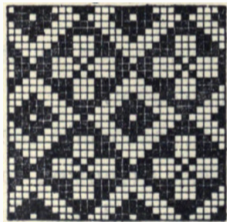
(8)



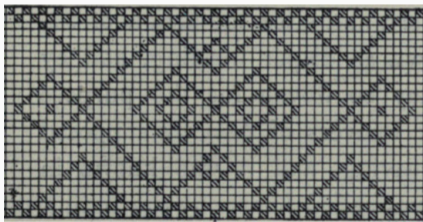
(14)

Tear de Jacquard

- Tramas complexas envolvem mais conjuntos de linhas ao invés de apenas dois
- Com um número não muito grande de conjuntos, é possível gerar vários padrões geométricos (repetitivos) diferentes

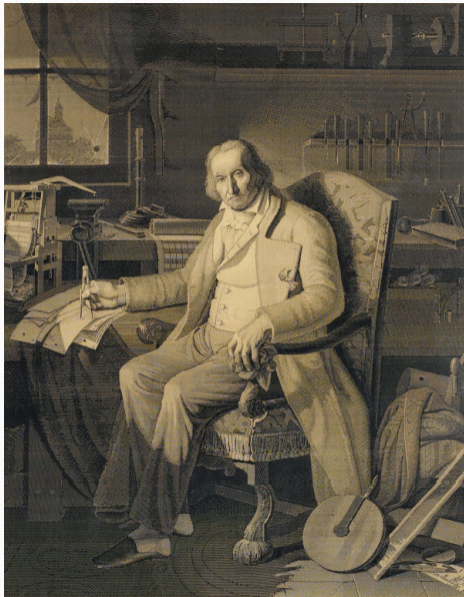


(8)



(14)

- Para gerar figuras realmente complexas, é necessário selecionar o movimento das linhas manualmente a cada passo, limitando o que é possível na prática





Tear de Jacquard

- No tear de Jacquard, não há conjuntos de linhas; cada linha é movida para cima ou para baixo individualmente, mas de maneira automatizada

Tear de Jacquard

- **No tear de Jacquard, não há conjuntos de linhas; cada linha é movida para cima ou para baixo individualmente, mas de maneira automatizada**
 - ▶ (cada linha está presa a um gancho que se move ou não a cada passo)

Tear de Jacquard

- **No tear de Jacquard, não há conjuntos de linhas; cada linha é movida para cima ou para baixo individualmente, mas de maneira automatizada**
 - ▶ (cada linha está presa a um gancho que se move ou não a cada passo)
- **O movimento das linhas a cada iteração é controlado por uma folha de papel com furos**

Tear de Jacquard

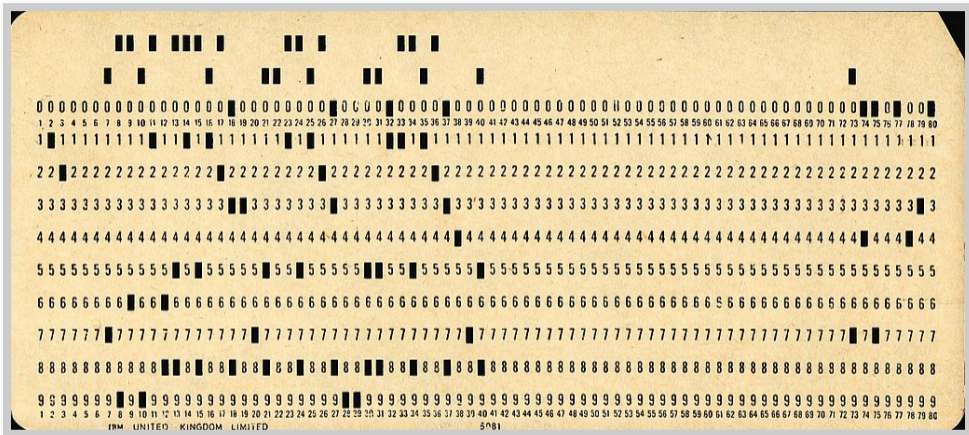
- **No tear de Jacquard, não há conjuntos de linhas; cada linha é movida para cima ou para baixo individualmente, mas de maneira automatizada**
 - ▶ (cada linha está presa a um gancho que se move ou não a cada passo)
- **O movimento das linhas a cada iteração é controlado por uma folha de papel com furos**
 - ▶ A folha de papel é pressionada contra um conjunto de hastes: onde não há furo, a haste é empurrada e move o gancho correspondente; onde há furo, a haste passa pelo furo e não se move

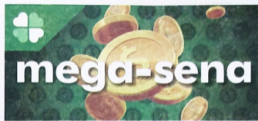
Tear de Jacquard

- **No tear de Jacquard, não há conjuntos de linhas; cada linha é movida para cima ou para baixo individualmente, mas de maneira automatizada**
 - ▶ (cada linha está presa a um gancho que se move ou não a cada passo)
- **O movimento das linhas a cada iteração é controlado por uma folha de papel com furos**
 - ▶ A folha de papel é pressionada contra um conjunto de hastes: onde não há furo, a haste é empurrada e move o gancho correspondente; onde há furo, a haste passa pelo furo e não se move
- **Uma mesma máquina é capaz de executar vários desenhos diferentes, além de reproduzir um mesmo desenho várias vezes — “hardware” vs “software”**



A ideia dos cartões perfurados influenciou Charles Babbage e Herman Hollerith, e se tornou o mecanismo padrão de entrada de dados nos computadores, permanecendo em uso até os anos 1980





VOCÊ PODE JOGAR MARCANDO EM 1, 2 OU 3 QUADROS ABAIXO:

<input type="checkbox"/>	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]
<input type="checkbox"/>	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
<input type="checkbox"/>	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
<input type="checkbox"/>	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
<input type="checkbox"/>	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]
<input type="checkbox"/>	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]

Para anular este jogo, marque ao lado: []

<input type="checkbox"/>	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]
<input type="checkbox"/>	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
<input type="checkbox"/>	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
<input type="checkbox"/>	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
<input type="checkbox"/>	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]
<input type="checkbox"/>	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]

Para anular este jogo, marque ao lado: []

<input type="checkbox"/>	[01]	[02]	[03]	[04]	[05]	[06]	[07]	[08]	[09]	[10]
<input type="checkbox"/>	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
<input type="checkbox"/>	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
<input type="checkbox"/>	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
<input type="checkbox"/>	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]
<input type="checkbox"/>	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]

Para anular este jogo, marque ao lado: []

Assinale quantos números você está marcando neste jogo:

[6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

SURPRESINHA - Aqui o sistema escolhe os números por você. Indique quantas apostas deseja fazer:

[1] [2] [3] [4] [5] [6] [7]

TEIMOSINHA - Escolha em quantos concursos você quer participar com este mesmo jogo (não é válido para Bôlão):

[2] [4] [8]

BOLÃO - Aqui você faz seu bolão de até 100 cotas. Assinale abaixo o nº de cotas:

[1] [2] [3] [4] [5] [6] [7] [8] [9] dezena

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] unidade

[100] Cota limite

CONFIRA O BILHETE IMPRESSO PELO TERMINAL. ELE É O ÚNICO COMPROVANTE DA APOSTA.



Preencha toda a área dos números escolhidos com caneta esferográfica azul ou preta.

**“I wish to God these calculations
had been executed by steam!”**

Charles Babbage

A máquina diferencial – 1/3

- **No séc XIX, as tabelas de logaritmos eram amplamente utilizadas, mas obviamente continham (muitos) erros**
 - ▶ (principalmente na preparação para a impressão)

A máquina diferencial – 1/3

- **No séc XIX, as tabelas de logaritmos eram amplamente utilizadas, mas obviamente continham (muitos) erros**
 - ▶ (principalmente na preparação para a impressão)
- **Babbage imaginou que seria possível construir uma máquina calculadora capaz de utilizar o método das diferenças divididas (apenas somas) para gerar essas tabelas de forma automática**

A máquina diferencial – 1/3

- **No séc XIX, as tabelas de logaritmos eram amplamente utilizadas, mas obviamente continham (muitos) erros**
 - (principalmente na preparação para a impressão)
- **Babbage imaginou que seria possível construir uma máquina calculadora capaz de utilizar o método das diferenças divididas (apenas somas) para gerar essas tabelas de forma automática**
- **Após a apresentação de um protótipo, o governo britânico financiou o projeto**

A máquina diferencial – 1/3

- **No séc XIX, as tabelas de logaritmos eram amplamente utilizadas, mas obviamente continham (muitos) erros**
 - (principalmente na preparação para a impressão)
- **Babbage imaginou que seria possível construir uma máquina calculadora capaz de utilizar o método das diferenças divididas (apenas somas) para gerar essas tabelas de forma automática**
- **Após a apresentação de um protótipo, o governo britânico financiou o projeto**
- **A primeira versão da máquina (incluindo uma impressora!) teria cerca de 25.000 peças, pesaria 4 toneladas e seria capaz de processar números com 20 dígitos de precisão, resolvendo polinômios de sexto grau (7 acumuladores)**

- O projeto acabou sendo abandonado

- **O projeto acabou sendo abandonado**
 - ▶ A fabricação das peças com a qualidade necessária era cara e difícil

- **O projeto acabou sendo abandonado**
 - ▶ A fabricação das peças com a qualidade necessária era cara e difícil
 - ▶ Babbage alterava constantemente o projeto, o que levou a vários desentendimentos com o fabricante

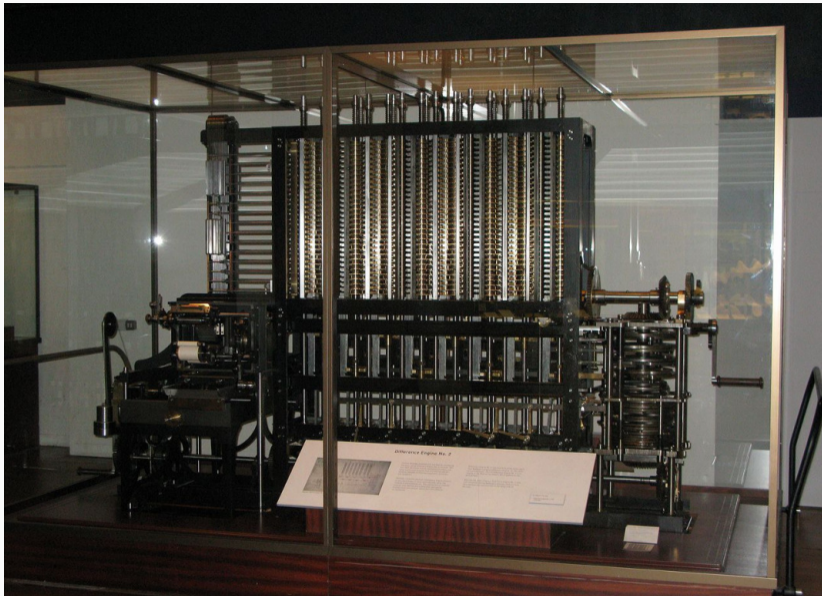
- **O projeto acabou sendo abandonado**
 - ▶ A fabricação das peças com a qualidade necessária era cara e difícil
 - ▶ Babbage alterava constantemente o projeto, o que levou a vários desentendimentos com o fabricante
 - ▶ Com isso, o custo se tornou proibitivo: em valores atuais, o governo britânico investiu cerca de US\$3 milhões (suficientes para comprar 22 locomotivas na época) e, após 20 anos, ainda não tinha algo funcional

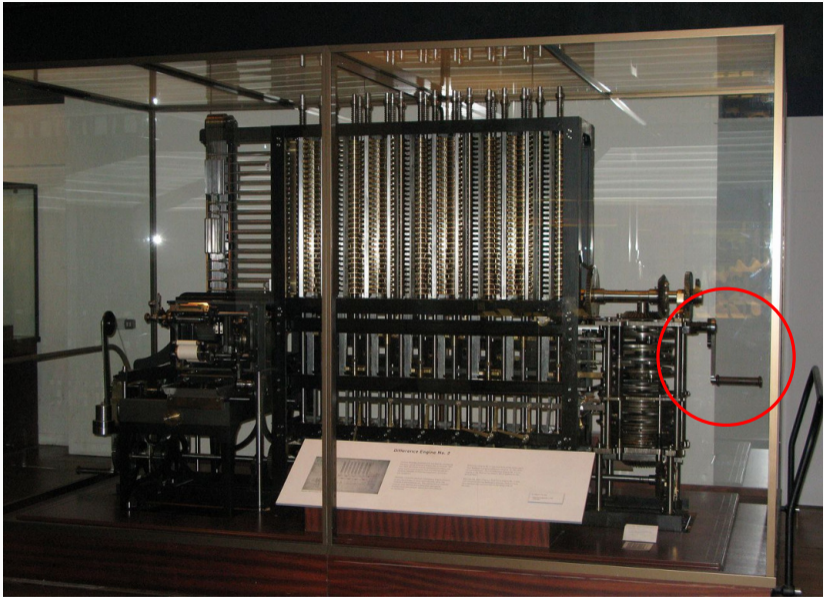
- **O projeto acabou sendo abandonado**
 - ▶ A fabricação das peças com a qualidade necessária era cara e difícil
 - ▶ Babbage alterava constantemente o projeto, o que levou a vários desentendimentos com o fabricante
 - ▶ Com isso, o custo se tornou proibitivo: em valores atuais, o governo britânico investiu cerca de US\$3 milhões (suficientes para comprar 22 locomotivas na época) e, após 20 anos, ainda não tinha algo funcional
 - ▶ Além disso, o próprio Babbage perdeu o interesse na máquina diferencial por causa da máquina analítica

- **Outras pessoas adotaram o design e construíram versões reduzidas da máquina, com menor precisão e capazes de lidar com polinômios de grau menor**

A máquina diferencial – 3/3

- **Outras pessoas adotaram o design e construíram versões reduzidas da máquina, com menor precisão e capazes de lidar com polinômios de grau menor**
- **Em 1991, o projeto revisado de Babbage, com apenas 8.000 peças, pesando 5 toneladas e precisão de 31 dígitos com polinômios de sétimo grau, foi executado (simulando os limites de precisão das peças no séc XIX) em comemoração ao aniversário de 200 anos de seu nascimento. Hoje existem duas máquinas diferenciais no mundo, que funcionam como esperado.**





A máquina analítica e Ada Lovelace – 1/2

- Durante o trabalho com a máquina diferencial, Babbage acabou por conceber um projeto muito mais sofisticado e ambicioso: a máquina analítica

A máquina analítica e Ada Lovelace — 1/2

- **Durante o trabalho com a máquina diferencial, Babbage acabou por conceber um projeto muito mais sofisticado e ambicioso: a máquina analítica**
 - ▶ (“analítica” porque dividia cada problema em uma sequência de passos pequenos)

A máquina analítica e Ada Lovelace — 1/2

- **Durante o trabalho com a máquina diferencial, Babbage acabou por conceber um projeto muito mais sofisticado e ambicioso: a máquina analítica**
 - ▶ (“analítica” porque dividia cada problema em uma sequência de passos pequenos)
- **Embora nunca tenha sido construída, a máquina analítica seria um computador Turing-completo**

A máquina analítica e Ada Lovelace — 1/2

- **Durante o trabalho com a máquina diferencial, Babbage acabou por conceber um projeto muito mais sofisticado e ambicioso: a máquina analítica**
 - ▶ (“analítica” porque dividia cada problema em uma sequência de passos pequenos)
- **Embora nunca tenha sido construída, a máquina analítica seria um computador Turing-completo**
- **Para a entrada de dados e de programas, a máquina utilizaria cartões de papel perfurado — Babbage foi fortemente influenciado pelo tear de Jacquard**

A máquina analítica e Ada Lovelace — 1/2

- **Durante o trabalho com a máquina diferencial, Babbage acabou por conceber um projeto muito mais sofisticado e ambicioso: a máquina analítica**
 - ▶ (“analítica” porque dividia cada problema em uma sequência de passos pequenos)
- **Embora nunca tenha sido construída, a máquina analítica seria um computador Turing-completo**
- **Para a entrada de dados e de programas, a máquina utilizaria cartões de papel perfurado — Babbage foi fortemente influenciado pelo tear de Jacquard**
- **Para a saída de dados, a máquina teria uma impressora e seria capaz de plotar gráficos, além de perfurar cartões de papel para serem lidos em outra execução**

A máquina analítica e Ada Lovelace – 2/2

- Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica

A máquina analítica e Ada Lovelace – 2/2

- Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica
- Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si

A máquina analítica e Ada Lovelace – 2/2

- Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica
- Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si
- Foi Ada quem se deu conta do imenso potencial da máquina analítica, a depender apenas dos programas que fossem criados para ela

A máquina analítica e Ada Lovelace – 2/2

- Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica
- Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si
- Foi Ada quem se deu conta do imenso potencial da máquina analítica, a depender apenas dos programas que fossem criados para ela
 - ▶ Algoritmos podem ser generalizados

A máquina analítica e Ada Lovelace – 2/2

- Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica
- Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si
- Foi Ada quem se deu conta do imenso potencial da máquina analítica, a depender apenas dos programas que fossem criados para ela
 - ▶ Algoritmos podem ser generalizados
 - ▶ É possível fazer atribuição de variáveis (não usar apenas entrada, saída e “acumuladores”)

A máquina analítica e Ada Lovelace – 2/2

- **Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica**
- **Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si**
- **Foi Ada quem se deu conta do imenso potencial da máquina analítica, a depender apenas dos programas que fossem criados para ela**
 - ▶ Algoritmos podem ser generalizados
 - ▶ É possível fazer atribuição de variáveis (não usar apenas entrada, saída e “acumuladores”)
 - ▶ É possível usar outros símbolos além de números, como notas musicais

A máquina analítica e Ada Lovelace – 2/2

- **Babbage conheceu Ada Lovelace, que se interessou grandemente pela máquina analítica**
- **Babbage criou esboços de programas para sua máquina, mas estava focado em cálculos matemáticos e na construção do engenho em si**
- **Foi Ada quem se deu conta do imenso potencial da máquina analítica, a depender apenas dos programas que fossem criados para ela**
 - ▶ Algoritmos podem ser generalizados
 - ▶ É possível fazer atribuição de variáveis (não usar apenas entrada, saída e “acumuladores”)
 - ▶ É possível usar outros símbolos além de números, como notas musicais
- **Seus escritos, em particular a primeira publicação de um algoritmo completo (cálculo de números de Bernoulli, baseado em uma ideia de Babbage) e a discussão sobre ele, fazem com que ela seja considerada a primeira programadora da história**

Hollerith e o censo americano

- Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática

Hollerith e o censo americano

- **Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática**
 - ▶ Os cartões eram lidos pressionando um conjunto de conectores com molas sobre eles; nos locais em que havia furos, o conector encostava em um contato de mercúrio, acionando um mecanismo elétrico que incrementava o contador correspondente

Hollerith e o censo americano

- **Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática**
 - Os cartões eram lidos pressionando um conjunto de conectores com molas sobre eles; nos locais em que havia furos, o conector encostava em um contato de mercúrio, acionando um mecanismo elétrico que incrementava o contador correspondente
- **A empresa desenvolveu variações do sistema e máquinas auxiliares para a realização de diversas tarefas**

Hollerith e o censo americano

- **Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática**
 - Os cartões eram lidos pressionando um conjunto de conectores com molas sobre eles; nos locais em que havia furos, o conector encostava em um contato de mercúrio, acionando um mecanismo elétrico que incrementava o contador correspondente
- **A empresa desenvolveu variações do sistema e máquinas auxiliares para a realização de diversas tarefas**
 - Estatísticas como “total de vendas por vendedor”, “total de vendas por produtos” etc.

Hollerith e o censo americano

- **Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática**
 - ▶ Os cartões eram lidos pressionando um conjunto de conectores com molas sobre eles; nos locais em que havia furos, o conector encostava em um contato de mercúrio, acionando um mecanismo elétrico que incrementava o contador correspondente
- **A empresa desenvolveu variações do sistema e máquinas auxiliares para a realização de diversas tarefas**
 - ▶ Estatísticas como “total de vendas por vendedor”, “total de vendas por produtos” etc.
 - ▶ Totalização de compras e vendas para faturamento e pagamento

Hollerith e o censo americano

- **Para acelerar a tabulação de dados do censo, Hollerith imaginou usar cartões perfurados pelos recenseadores que depois poderiam ser lidos de maneira automática**
 - ▶ Os cartões eram lidos pressionando um conjunto de conectores com molas sobre eles; nos locais em que havia furos, o conector encostava em um contato de mercúrio, acionando um mecanismo elétrico que incrementava o contador correspondente
- **A empresa desenvolveu variações do sistema e máquinas auxiliares para a realização de diversas tarefas**
 - ▶ Estatísticas como “total de vendas por vendedor”, “total de vendas por produtos” etc.
 - ▶ Totalização de compras e vendas para faturamento e pagamento
 - ▶ Em conjunto com os cartões de ponto, cálculo de horas de trabalho e correspondente pagamento de funcionários (“Hollerith”)

Expansão dos “computadores” mecânicos

- Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos

Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...

Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...
 - ▶ Versões reduzidas da máquina diferencial (apenas 3 ou 4 unidades)

Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...
 - ▶ Versões reduzidas da máquina diferencial (apenas 3 ou 4 unidades)
 - ▶ Diversos equipamentos baseados na ideia de Hollerith para contabilidade empresarial

Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...
 - ▶ Versões reduzidas da máquina diferencial (apenas 3 ou 4 unidades)
 - ▶ Diversos equipamentos baseados na ideia de Hollerith para contabilidade empresarial
 - ▶ Vários modelos de máquinas enigma, usadas para criptografar as comunicações das forças nazistas durante a guerra

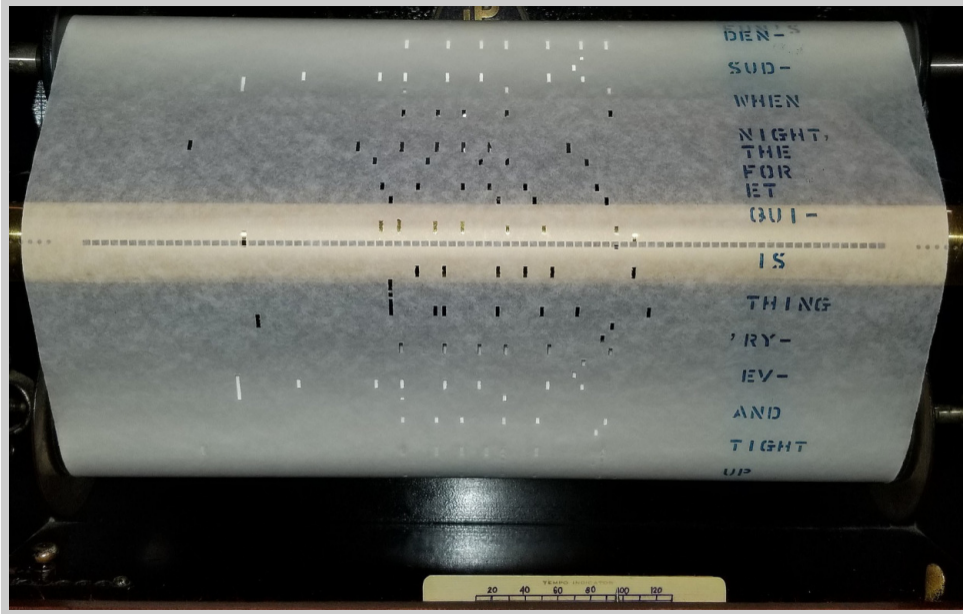
Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...
 - ▶ Versões reduzidas da máquina diferencial (apenas 3 ou 4 unidades)
 - ▶ Diversos equipamentos baseados na ideia de Hollerith para contabilidade empresarial
 - ▶ Vários modelos de máquinas enigma, usadas para criptografar as comunicações das forças nazistas durante a guerra
 - ▶ A “bomba kryptologiczna” polonesa e a “bombe” inglesa, usadas para decifrar mensagens criadas pela enigma

Expansão dos “computadores” mecânicos

- **Nos séculos XIX–XX, alguns “computadores” mecânicos e eletromecânicos foram de fato construídos**
 - ▶ Calculadoras mecânicas, caixas registradoras, relógios de ponto...
 - ▶ Versões reduzidas da máquina diferencial (apenas 3 ou 4 unidades)
 - ▶ Diversos equipamentos baseados na ideia de Hollerith para contabilidade empresarial
 - ▶ Vários modelos de máquinas enigma, usadas para criptografar as comunicações das forças nazistas durante a guerra
 - ▶ A “bomba kryptologiczna” polonesa e a “bombe” inglesa, usadas para decifrar mensagens criadas pela enigma
 - ▶ A pianola também foi concebida nesse período, e também era controlada por uma fita de papel perfurado





De Hollerith à IBM

- A empresa de Hollerith teve grande sucesso e acabou fundindo-se com outras três (fabricantes de relógios de ponto, calculadoras mecânicas e balanças “inteligentes”), dando origem à IBM algumas décadas depois

De Hollerith à IBM

- A empresa de Hollerith teve grande sucesso e acabou fundindo-se com outras três (fabricantes de relógios de ponto, calculadoras mecânicas e balanças “inteligentes”), dando origem à IBM algumas décadas depois
- Ao longo do século XX, a tecnologia usada foi aprimorada, chegando à adoção de sistemas eletrônicos nos anos 1940

De Hollerith à IBM

- A empresa de Hollerith teve grande sucesso e acabou fundindo-se com outras três (fabricantes de relógios de ponto, calculadoras mecânicas e balanças “inteligentes”), dando origem à IBM algumas décadas depois
- Ao longo do século XX, a tecnologia usada foi aprimorada, chegando à adoção de sistemas eletrônicos nos anos 1940
- A IBM esteve fortemente envolvida no desenvolvimento do Harvard Mark I (1944), o que lhe permitiu desenvolver seus próprios produtos: SSEC (1948, essencialmente uma jogada de marketing) e IBM 701/702/650 (1952/53/54)

De Hollerith à IBM

- A empresa de Hollerith teve grande sucesso e acabou fundindo-se com outras três (fabricantes de relógios de ponto, calculadoras mecânicas e balanças “inteligentes”), dando origem à IBM algumas décadas depois
- Ao longo do século XX, a tecnologia usada foi aprimorada, chegando à adoção de sistemas eletrônicos nos anos 1940
- A IBM esteve fortemente envolvida no desenvolvimento do Harvard Mark I (1944), o que lhe permitiu desenvolver seus próprios produtos: SSEC (1948, essencialmente uma jogada de marketing) e IBM 701/702/650 (1952/53/54)
- Até os anos 1980, a IBM era maior que todos seus concorrentes juntos → **IBM e os sete anos**