

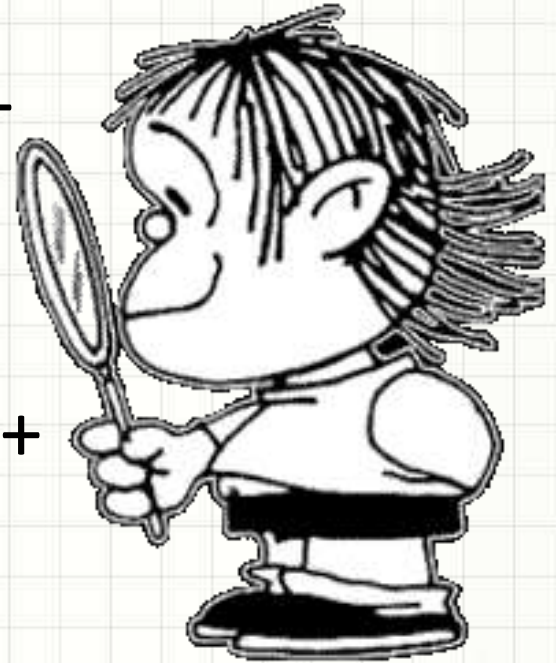
LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA AMBIENTE DE PROGRAMAÇÃO

Prof. Dr. Daniel Caetano

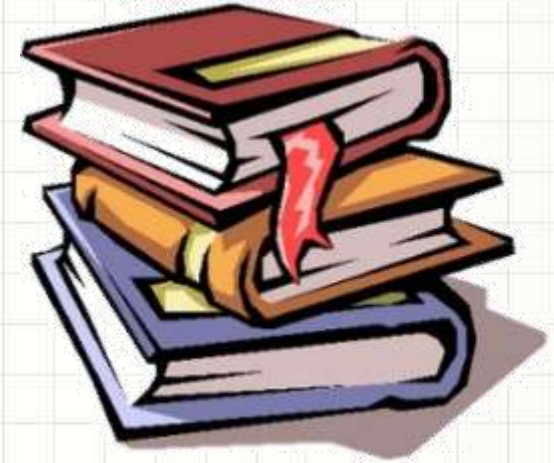
2012 - 2

Objetivos

- Entender a utilidade do resto de divisão
- Conhecer as funções matemáticas prontas do C/C++
- Capacitar o aluno para criar algoritmos sequenciais com funções matemáticas em C/C++
- **LISTA 1**



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>
(Aula 6)

Apresentação

<http://www.caetano.eng.br/>
(Aula 6)

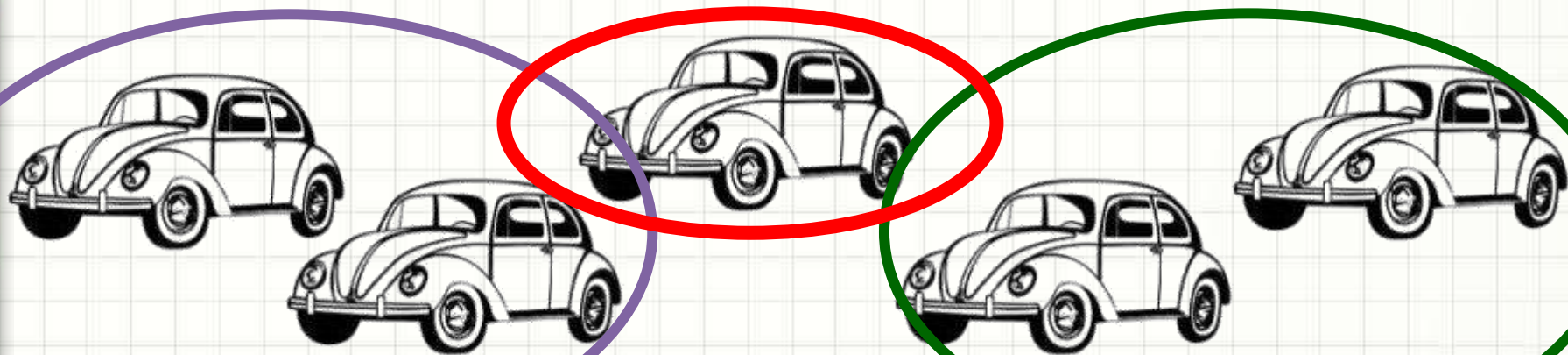
Material Didático

Lógica de Programação – Fundamentos da
Programação de Computadores, páginas 7 a 47.



RESTO DE DIVISÃO

Resto de Divisão



Número Par ou Ímpar?

- Como determinar se um número é par?
- Par: divisível por dois
- O que significa ser divisível por 2?
- Significa que o resto da divisão por 2 é 0!

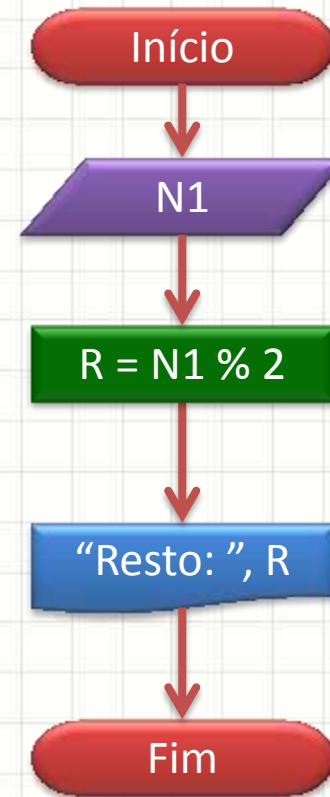
- Vamos experimentar:
 - Algoritmo que imprime “0” se o número é **par** e “1” se o número é **ímpar**

Verificando Paridade

- Linguagem Natural

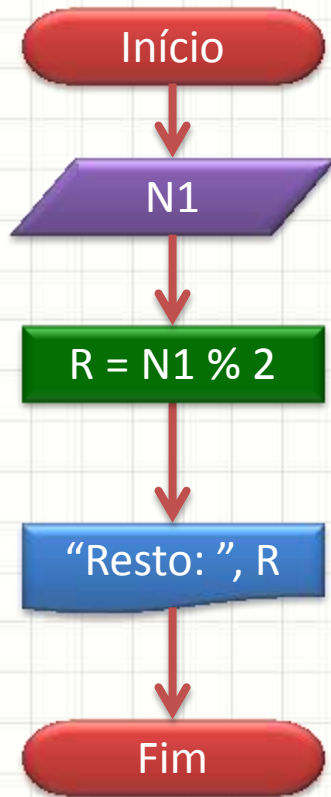
1. Leia um número
2. Calcule o resto da divisão por 2
3. Imprima o resto

- Fluxograma



Verificando Paridade

- Fluxograma



- Portugol

Algoritmo "Calcula Paridade"

Var

INTEIRO : N1, R

Inicio

Escreva("Digite Um Número:")

Leia(N1)

$R \leftarrow N1 \% 2$

Escreva("Resto:", R)

FimAlgoritmo

Verificando Paridade

Algoritmo “Calcula Paridade”

Var

INTEIRO : N1, R

Inicio

Escreva(“Digite Um Número:”)

Leia(N1)

$R \leftarrow N1 \% 2$

Escreva(“Resto:”, R)

FimAlgoritmo

Verificando Paridade

- Portugol

Algoritmo “Calcula Paridade”

Var

INTEIRO : N1, R

Inicio

Escreva(“Digite Um Número:”)

Leia(N1)

$R \leftarrow N1 \% 2$

Escreva(“Resto:”, R)

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void) {
```

```
    int N1, R;
```

```
    cout << “Digite Um Número:”;
```

```
    cin >> N1;
```

```
     $R = N1 \% 2;$ 
```

```
    cout << “Resto: ” << R;
```

```
}
```

Verificando Paridade

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void) {
```

```
    int N1, R;
```

```
    cout << "Digite Um Número:";
```

```
    cin >> N1;
```

```
    R = N1 % 2;
```

```
    cout << "Resto: " << R;
```

```
}
```

Verificando Paridade

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{  
    int
```

```
    cin
```

```
    int
```

```
    R = N1 % 2;
```

```
    cout << "Resto: " << R;
```

```
}
```

Como imprimir "Par" se o número é par e "Ímpar" se o número é ímpar?

Conversão de Segundos para H:M:S

- Convertendo 54.346 segundos em
 - Horas, Minutos e Segundos

54.346

H : M : S

Conversão de Segundos para H:M:S

- Convertendo 54.346s \rightarrow H : M : S
- 1h \rightarrow 3.600s
- Quantas horas tem em 54.346s?

$$54.346 / 3.600 = 15,09611111... \text{ horas}$$

15 : M : S

- 15 horas e “uns quebrados”...
- Quantos segundos sobraram?

Conversão de Segundos para H:M:S

- Convertendo 54.346s \rightarrow 15 : M : S
- 1h \rightarrow 3.600s
- Quantos segs. não completam 1h em 54.346s?
 $54.346 \% 3.600 = 346... \text{ segundos}$

15 : M : 346 ?

Conversão de Segundos para H:M:S

- Convertendo 54.346s → 15 : M : S
- 1min → 60s
- Quantos mins. tem em 346s?

$$346 / 60 = 5,7666666... \text{ minutos}$$

15 : 05 : S

- 5 minutos e “uns quebrados”...
- Quantos segundos sobraram?

Conversão de Segundos para H:M:S

- Convertendo 54.346s → 15 : 05 : S
- 1min → 60s
- Quantos segs. não completam 1min. em 346s?
 $346 \% 60 = 46... \text{ segundos}$

15 : 05 : 46

Conversão de Segundos para H:M:S

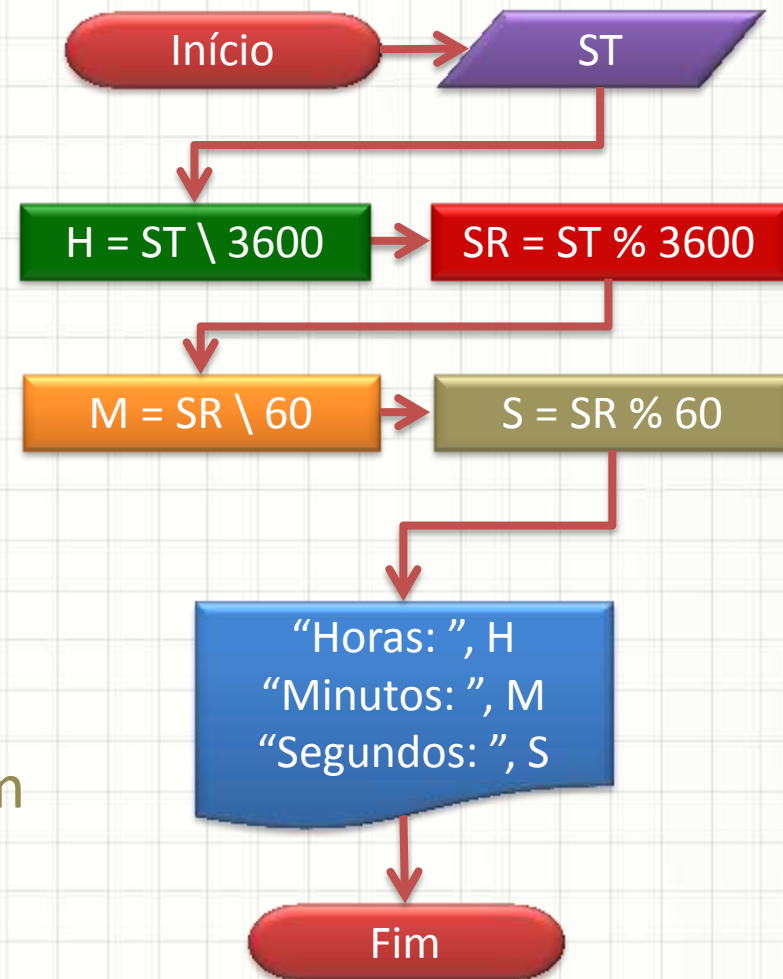
- Resumindo
 - $54.346 / 3.600 = \mathbf{15,09611111...}$ horas
 - $54.346 \% 3.600 = 346$ segundos (sobram)
 - $346 / 60 = \mathbf{5,76666666...}$ minutos
 - $346 \% 60 = \mathbf{46}$ segundos (sobram)
- Assim:
 - $54.346s = 15h, 5min, 46s$
- Como fazer um programa que calcule isso?

Conversão de Segundos para H:M:S

- Linguagem Natural

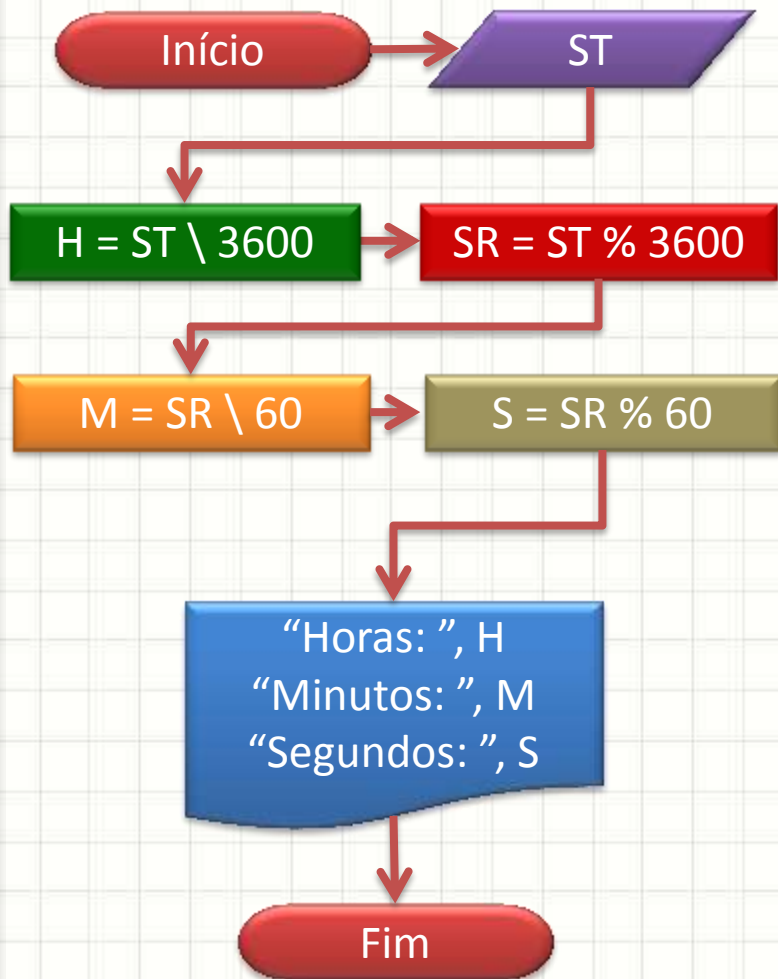
1. Leia o número de segundos totais
2. Calcule o número de horas dividindo por 3600 (divisão inteira)
3. Calcule os segundos restantes com o resto de divisão por 3600
4. Calcule o número de minutos dividindo o resto anterior por 60 (divisão inteira)
5. Calcule os segundos finais com o resto de divisão por 60
6. Imprima o número de horas, minutos e segundos

- Fluxograma



Conversão de Segundos para H:M:S

- Fluxograma



- Portugol

Algoritmo “Número de Semanas”

Var

INTEIRO: ST, H, SR, M, S

Início

Escreva(“Quantos segundos? ”)

Leia(ST)

$H \leftarrow ST \ \backslash \ 3600$

$SR \leftarrow ST \ \% \ 3600$

$M \leftarrow SR \ \backslash \ 60$

$S \leftarrow SR \ \% \ 60$

Escreval(“Horas: ”, H)

Escreval(“Minutos: ”, M)

Escreval(“Segundos: ”, S)

FimAlgoritmo

Conversão de Segundos para H:M:S

Algoritmo “Número de Semanas”

Var

INTEIRO: ST, H, SR, M, S

Início

Escreva(“Quantos segundos? ”)

Leia(ST)

$H \leftarrow ST \setminus 3600$

$SR \leftarrow ST \% 3600$

$M \leftarrow SR \setminus 60$

$S \leftarrow SR \% 60$

Escreval(“Horas: ”, H)

Escreval(“Minutos: ”, M)

Escreval(“Segundos: ”, S)

FimAlgoritmo

Conversão de Segundos para H:M:S

- Portugol

Algoritmo “Número de Semanas”

Var

INTEIRO: ST, H, SR, M, S

Início

Escreva(“Quantos segundos? ”)

Leia(ST)

$H \leftarrow ST \setminus 3600$

$SR \leftarrow ST \% 3600$

$M \leftarrow SR \setminus 60$

$S \leftarrow SR \% 60$

Escreval(“Horas: ”, H)

Escreval(“Minutos: ”, M)

Escreval(“Segundos: ”, S)

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void) {
```

```
    int ST, H, SR, M, S;
```

```
    cout << “Quantos segundos? ”;
```

```
    cin >> ST;
```

```
    H = ST / 3600;
```

```
    SR = ST % 3600;
```

```
    M = SR / 60;
```

```
    S = SR % 60;
```

```
    cout << “Horas: ” << H << endl;
```

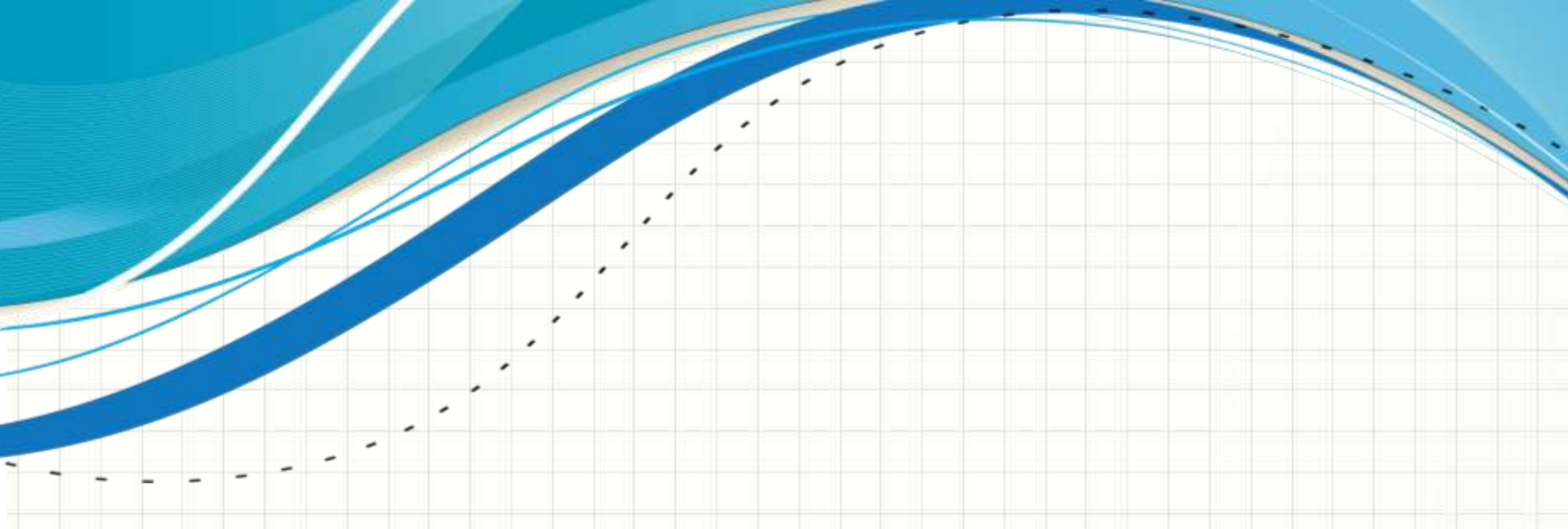
```
    cout << “Minutos: ” << M << endl;
```

```
    cout << “Segundos: ” << S << endl;
```

```
}
```

Conversão de Segundos para H:M:S

```
#include <iostream>
using namespace std;
int main(void) {
    int ST, H, SR, M, S;
    cout << "Quantos segundos? ";
    cin >> ST;
    H = ST / 3600;
    SR = ST % 3600;
    M = SR / 60;
    S = SR % 60;
    cout << "Horas: " << H << endl;
    cout << "Minutos: " << M << endl;
    cout << "Segundos: " << S << endl;
}
```



FUNÇÕES MATEMÁTICAS

Funções Matemáticas

- Que bando de cálculo “fuleiro”!
 - Onde estão as contas complexas?
 - Cadê o logaritmo, a raiz quadrada e outros?
- Esses “caras” são chamados de **funções**
- Existem várias funções prontas no C/C++

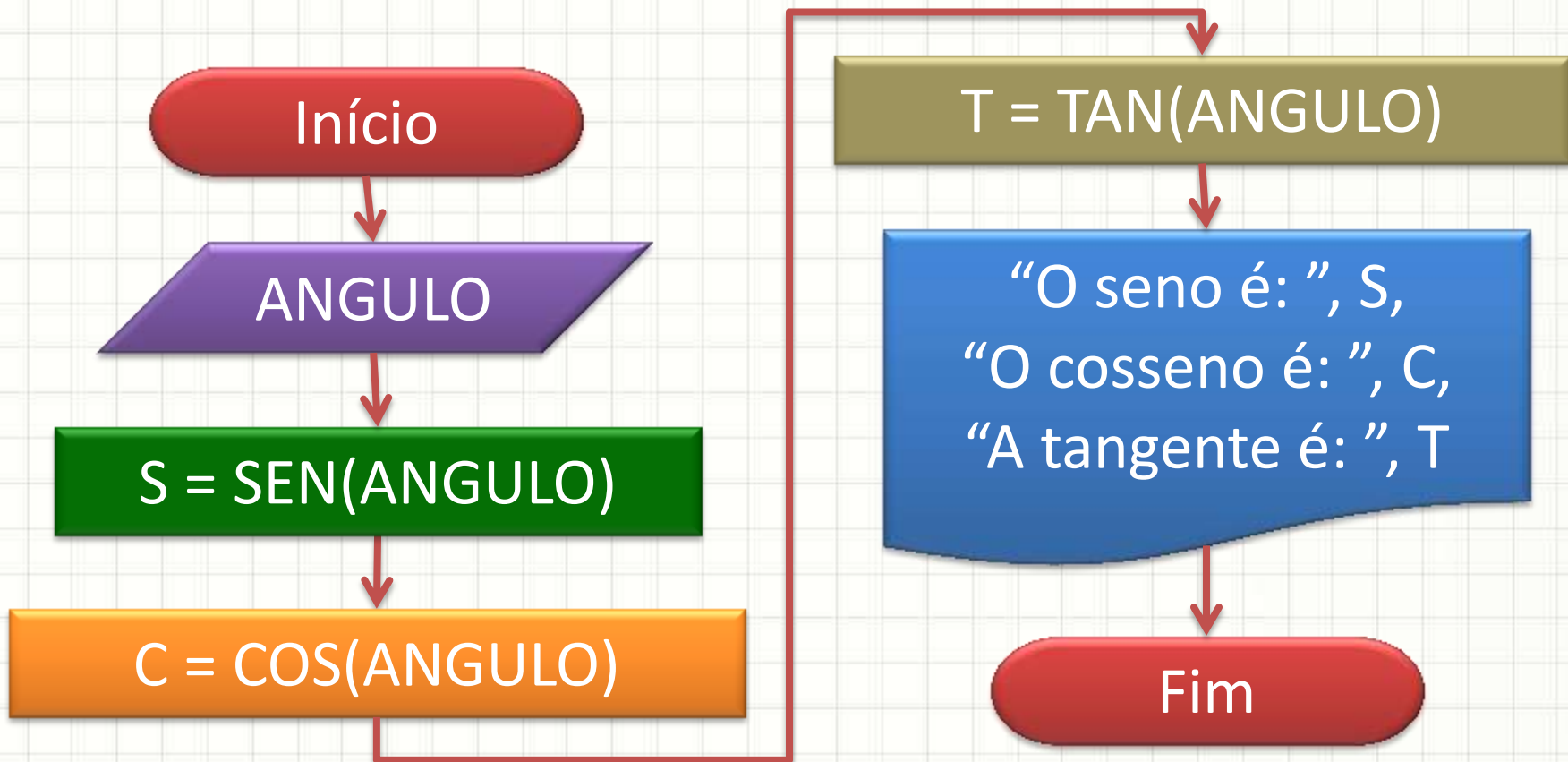
Funções Matemáticas

Portugol	C/C++	Função
-	abs(x)	Devolve o valor absoluto (sem sinal) de x
-	ceil(x)	Devolve x arredondado para cima
-	floor(x)	Devolve x arredondado para baixo
x^y	pow(x,y)	Devolve o valor de x^y
-	exp(x)	Devolve o valor de e^x
raizq(x)	sqrt(x)	Devolve o valor da raiz quadrada de x (SQure Root)
Pi	M_PI	Representa o valor de PI (3,141592...)
sen(x)	sin(x)	Devolve o valor do seno de x (radianos)
cos(x)	cos(x)	Devolve o valor do cosseno de x (radianos)
tan(x)	tan(x)	Devolve o valor da tangente de x (radianos)
-	asin(x)	Devolve o valor do arco-seno de x (arco em radianos)
-	acos(x)	Devolve o valor do arco-cosseno de x (arco em radianos)
-	atan(x)	Devolve o valor da arco-tangente de x (arco em radianos)

Senos, Cossenos e Tangentes

- Como um exemplo, vamos calcular senos, cossenos e tangentes
1. Vamos ler um ângulo
 2. Vamos calcular os valores
 3. Vamos imprimir os valores

Senos, Cossenos e Tangentes



Senos, Cossenos e Tangentes - P

Algoritmo “Senos, cossenos e tangente”

Var

REAL: ANGULO, S, C, T

Inicio

Escreva(“Digite um ângulo – 0 a $2 \cdot \text{PI}$: ”);

Leia(ANGULO)

$S \leftarrow \text{sen}(\text{ANGULO})$

$C \leftarrow \text{cos}(\text{ANGULO})$

$T \leftarrow \text{tan}(\text{ANGULO})$

Escreval(“Seno: ”, S)

Escreval(“Cosseno: ”, C)

Escreval(“Tangente: ”, T)

FimAlgoritmo

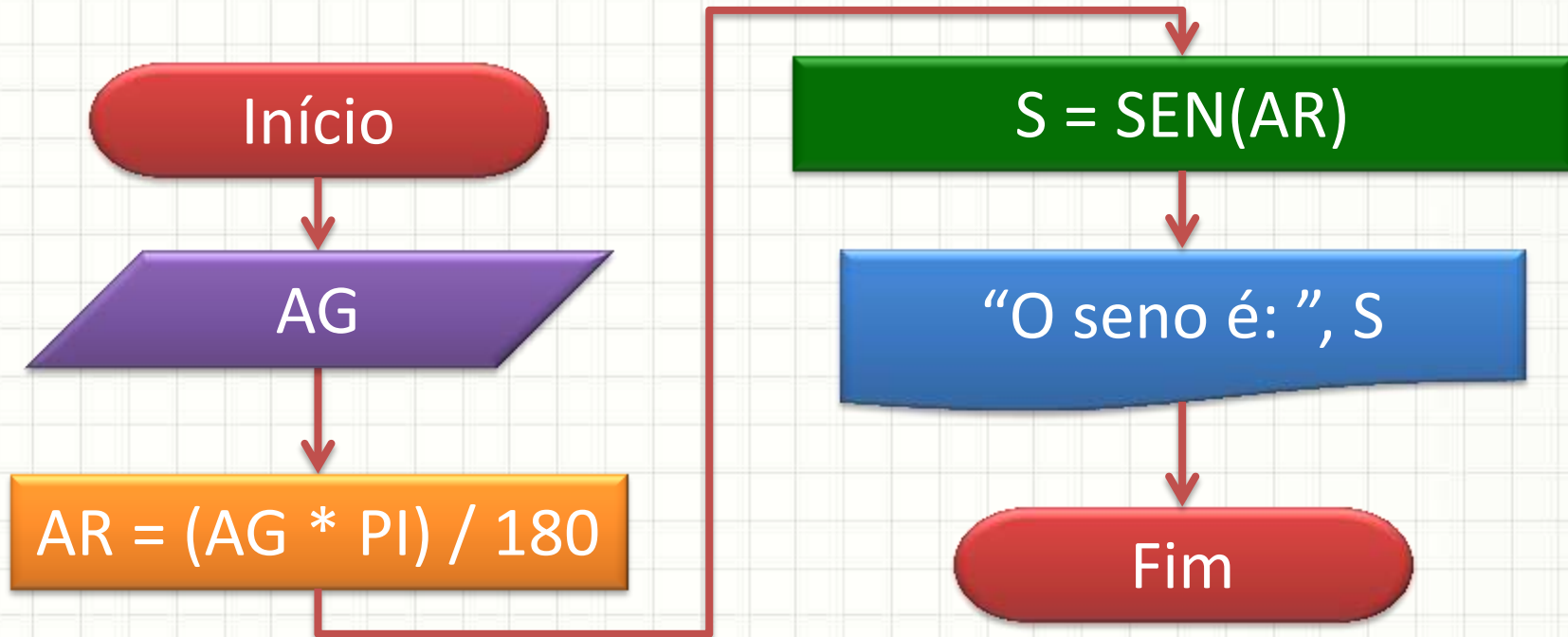
Senos, Cossenos e Tangentes - C

```
#include <iostream>
#include <math.h>
using namespace std;
int main(void) {
    float ANGULO, S, C, T;
    cout << "Digite um ângulo – 0 a 2*PI: ";
    cin >> ANGULO;
    S = sin(ANGULO);
    C = cos(ANGULO);
    T = tan(ANGULO);
    cout << "Seno: " << S << endl;
    cout << "Cosseno: " << C << endl;
    cout << "Tangente: " << T << endl;
}
```

Senos, Cossenos e Tangentes

- E se quisermos ler o ângulo em GRAUS?
 1. Vamos ler um ângulo (em graus)
 - 2. Vamos convertê-lo em radianos**
 3. Vamos calcular os valores
 4. Vamos imprimir os valores
- $ANGULO = (ANGULO_GRAUS * PI) / 180$

Senos, Cossenos e Tangentes



Senos, Cossenos e Tangentes - P

Algoritmo “Ângulo em graus”

Var

REAL : AG, AR, S

Inicio

Escreva(“Digite um ângulo – 0 a 360: ”);

Leia(AG)

$AR \leftarrow (AG * PI) / 180$

$S \leftarrow \text{sen}(AR)$

Escreval(“O seno é: ”, S)

FimAlgoritmo

Senos, Cossenos e Tangentes - P

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
int main(void) {
```

```
    float AG, AR, PI, S;
```

```
    cout << "Digite um ângulo – 0 a 360: ";
```

```
    cin >> AG;
```

```
    AR = (AG * M_PI) / 180.0;
```

```
    S = sin(AR);
```

```
    cout << "O seno é: " << S << endl;
```

```
}
```



PARA PENSAR:

ARREDONDAMENTO

Função de Arredondamento

- Ainda que o Portugol não forneça uma função de arredondamento, ela existe no C/C++

float valor, arredondado;

valor = 1.55;

arredondado = **round**(valor);

- Qual o valor de **arredondado**?
- Arredondamento em C: funções limitadas!

Arredondando Números - C

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
int main(void) {
```

```
    float NUM, ARRED;
```

```
    cout << "Digite um número fracionário: ";
```

```
    cin >> NUM;
```

```
    ARRED = round(NUM);
```

```
    cout << "Arredondado para: " << ARRED << endl;
```

```
    getchar();
```

```
}
```

Função de Arredondamento

- **floor** arredonda para o inteiro anterior
- **ceil** arredonda para o próximo inteiro
- **round** arredonda para inteiro mais próximo

- Para pensar:
 - Como fazer para arredondar com 1 casa decimal?
 - E com 2 casas decimais?
 - E com 3?



CONCLUSÕES

Resumo

- O uso de resto de divisão é útil para fracionar números em unidades menores e para verificar divisibilidade
- As bibliotecas do C/C++ fornecem uma porção de cálculos matemáticos prontos para serem usados
- **TAREFA!**
 - Lista de Exercícios 1

Próxima Aula



- Como posso reaproveitar algoritmos complicados?
 - Vou precisar reprogramar todas as vezes?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**