



PROGRAMAÇÃO DE COMPUTADORES I

SEXTA PARTE

CAPÍTULO III
ESTRUTURAS DE
CONTROLE DE FLUXO

- PARTE 2:
POR REPETIÇÃO:
- ENQUANTO
 - REPITA ... ENQUANTO

Programação de Computadores I

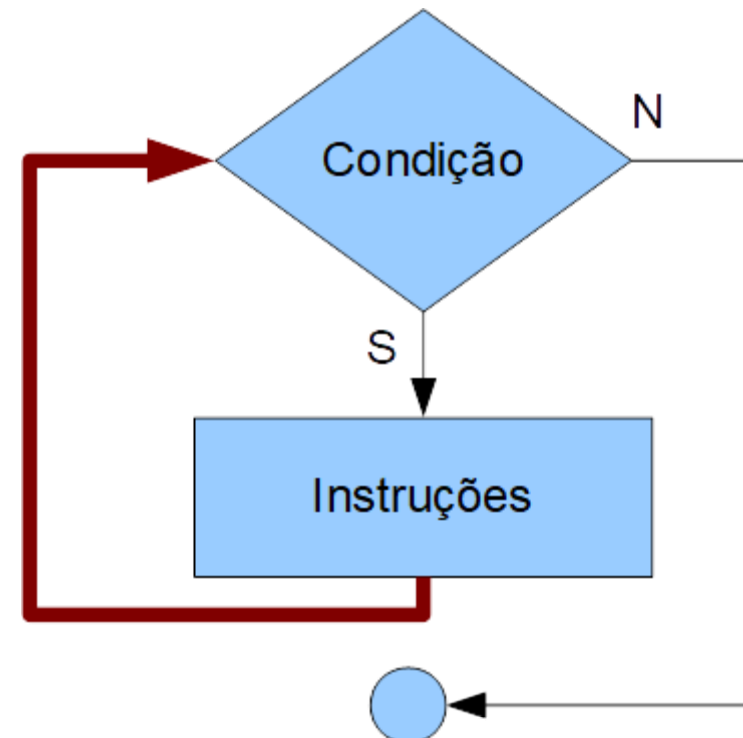
prof. Marco Villaça

Recapitulando

- Até agora, nos programas feitos:
 - ✓ a execução foi sequencial ou
 - ✓ sequencial com desvios por condição:
 - `if`
 - `if else`
 - `if else if else`

Estruturas de repetição

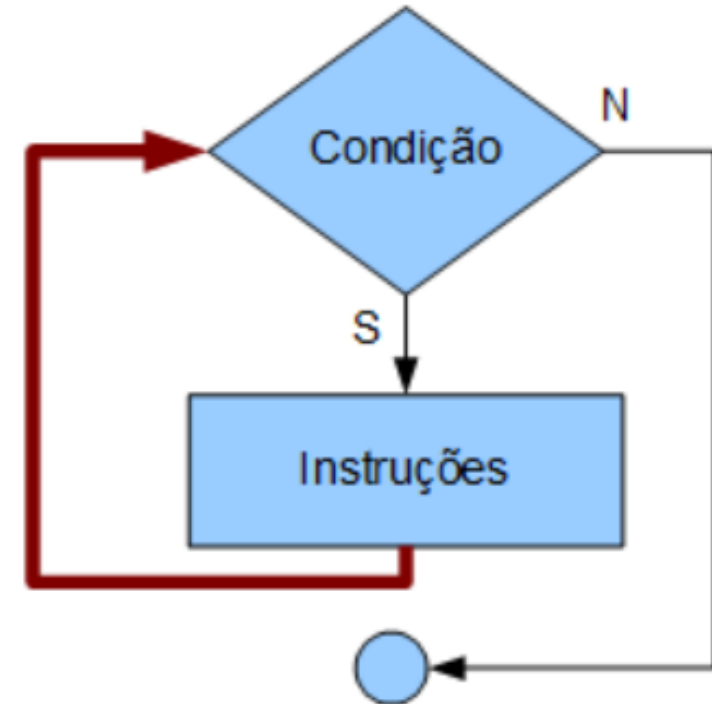
- Vamos passa a usar estruturas de repetição (loop ou laço)
- Três tipos
 - Enquanto (`while`)
 - Repita...enquanto (`do...while`)
 - Para (`for`)



Repetição com enquanto

- Teste lógico no início do laço
 - seta indica retorno para o bloco de teste
- Executa enquanto a condição for verdadeira

FLUXOGRAMA



Repetição com enquanto

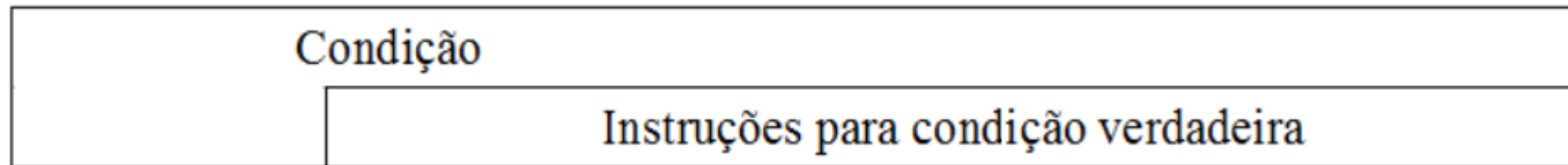
- Português estruturado

enquanto (<condição>) faça

 <instruções para condição verdadeira>

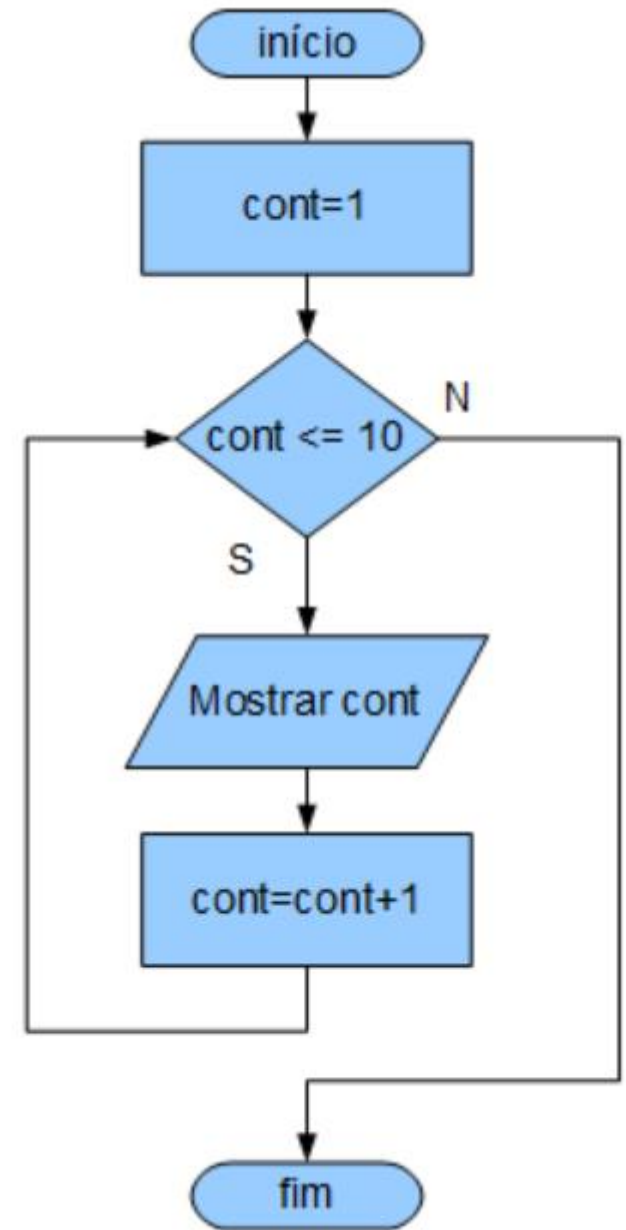
fim_enquanto

- Diagrama de Chapin



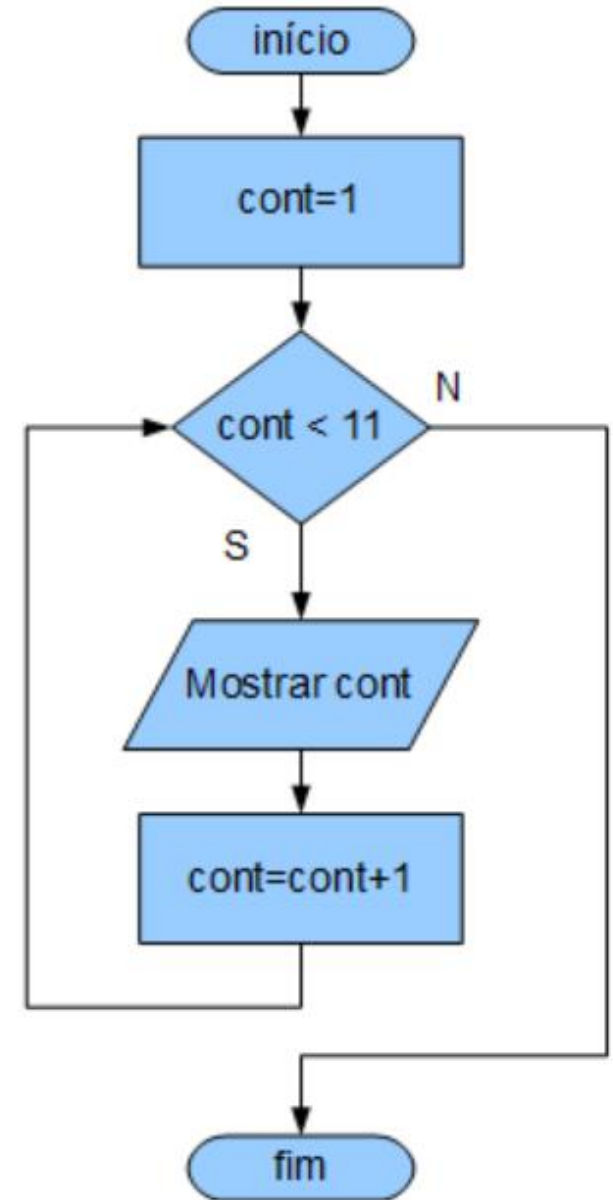
Repetição com enquanto

- Mostrar os números de 1 a 10 na tela



Repetição com enquanto

- Mostrar os números de 1 a 10 na tela



Repetição com enquanto em C

Comando `while`

- `while` (condição de teste)
 comando;
- Executa o comando (ou um bloco destes) **ENQUANTO** a condição for verdadeira (diferente de 0)

*/*imprimir números de 0 até 9 */*

```
int num=0;
while (num<10) {
    printf("%d\n", num);
    num++;
}
```

Repetição com enquanto no Scilab

Comando `while`

- **while** condição de teste **do** //ou **then** ou ,
comandos
end

- Exemplo:

```
k=1;  
while k<10 do  
    disp(k);  
    k = k + 1;  
end
```



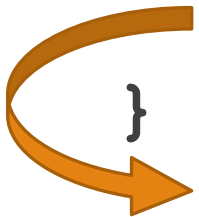
EXEMPLO 1

ESCREVA UM PROGRAMA QUE APRESENTE OS NÚMERO MÚLTIPLOS DE 3 E 5 ENTRE 0 E 100.

Saindo de um laço prematuramente em C

Comando break

```
while (1)           // Laço infinito
{
...                // Seu programa
    fflush(stdin);
    printf ("Sair? S ou N\n");
    resp = getchar( );
    if (resp == 'S' || resp == 's' )
        break;
```



Saindo de um laço prematuramente no Scilab

Comando break

```
k=1;
while 1 do
    disp(k);
    k = k + 1;
    a=input('Sair <s> ou <n>', 's');
    if a=='s' | a=='S' then
        break;
    end
end
```



Usando um laço para forçar a entrada correta de dados em C

```
while (1)
{
    printf( "Entre com sua idade: " );
    if (scanf ("%d", &age) == 1)
        break;
    fflush (stdin);
    printf( "\nErro ao ler a idade, tente de novo.\n" );
}
```

- A função `scanf` retorna o número de dados lidos.


EXEMPLO 2

- Escreva um programa que solicite duas notas de um aluno e imprima a média.
- Caso a média seja maior que 6, informar que o aluno está dispensado da recuperação.
- Após apresentar o resultado, perguntar se o usuário deseja sair ou continuar.



Passando prematuramente para a próxima iteração em C


Comando `continue`



```
while (1)                // Laço infinito
{
    printf (“Sexo? M ou F, \n”);
    fflush(stdin);
    r = getchar( );
    if (r != 'M' && r != 'm' && r != 'F' && r != 'f' )
        continue;
    ...
}
```


Passando prematuramente para a próxima iteração no Scilab

Comando `continue`



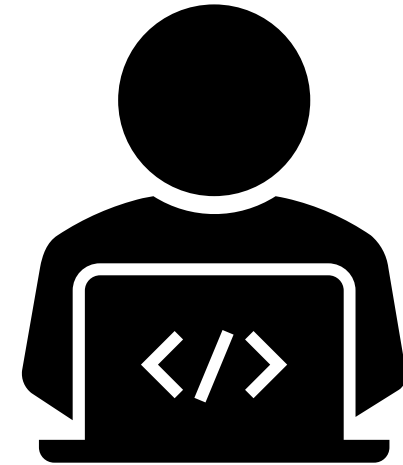
```
while 1 do  
    a=input('sexo <M> ou <F>? ', 's');  
    if a~='M' & a~='F' & a~='m' & a~='f' then  
        continue;  
    end  
    ...  
end
```

EXEMPLO 3

- Chico tem 1,50 metro e cresce 1,8 centímetros por ano, enquanto Zé tem 1,42 metro e cresce 3,2 centímetros por ano. Elabore um programa que calcule e imprima quantos anos serão necessários para que Zé seja maior que Chico.

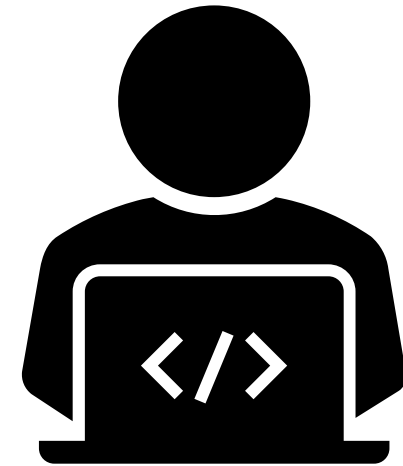


- Faça programas (usando um laço) que mostrem na tela as sequências
 - ✓ a. 1 3 5 7 9 11 13
 - ✓ b. 1 3 5 7 ... $N_{\text{máx}}$... 7 5 3 1
 - onde um $N_{\text{máx}}$ ímpar é solicitado ao usuário.
 - O programa só imprime a sequência se um número ímpar for digitado



Exercício 1

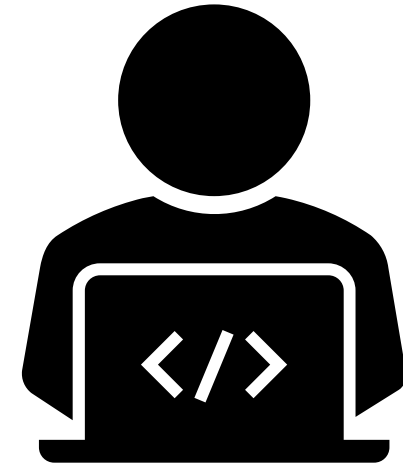
- Modifique o programa de treinamento (Exercício 1 da quinta semana) para:
 - Caso o usuário informe o sexo errado, ele solicita novamente;
 - Após informar o tempo de treinamento, solicita se o usuário deseja sair ou continuar



Exercício 2

- Faça um programa que
 - Peça um número ao usuário
 - Mostre na tela a tabuada desse número

Exemplo de uso
Digite um número: 3
A tabuada do 3 é
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
Fim do programa

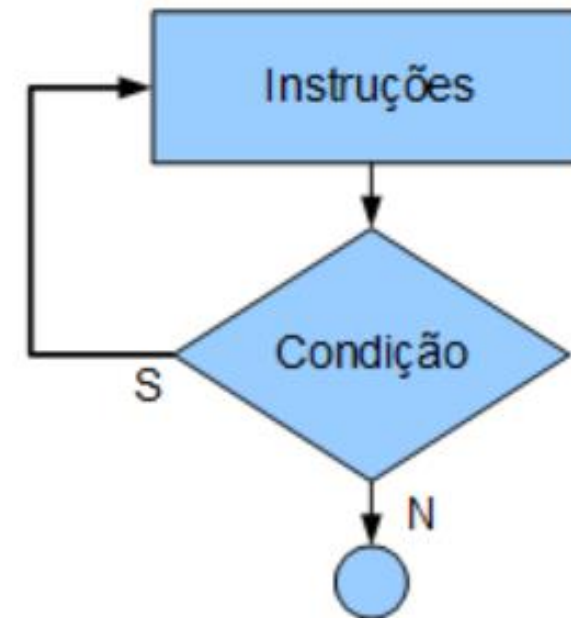


Exercício 3

Repita ... enquanto (do ... while)

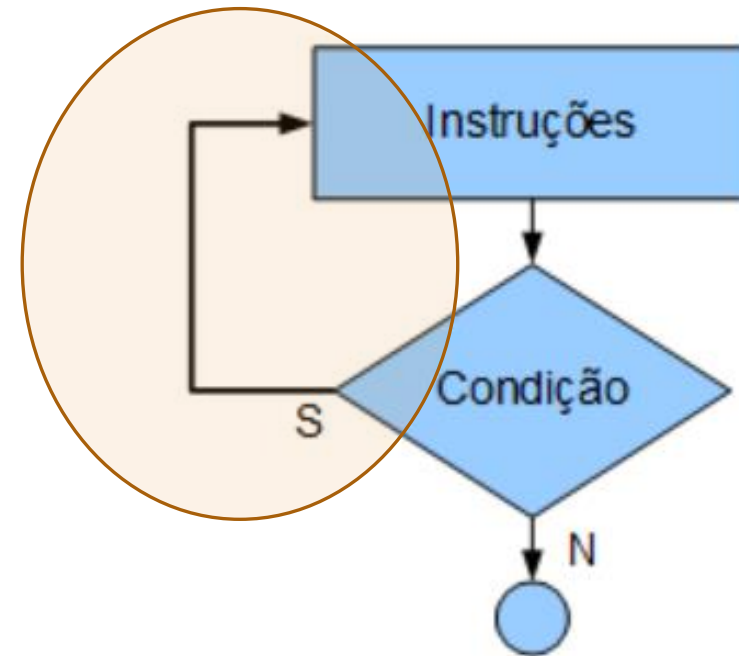
- Teste lógico no **final** do laço
 - Seta sai da decisão e vai para uma instrução
- Executa instruções dentro do laço **pelo menos uma vez**

FLUXOGRAMA



Repita ... enquanto (do ... while)

- Importante
 - seta tem que partir da condição **verdadeira**



Repita ... enquanto (do ... while)

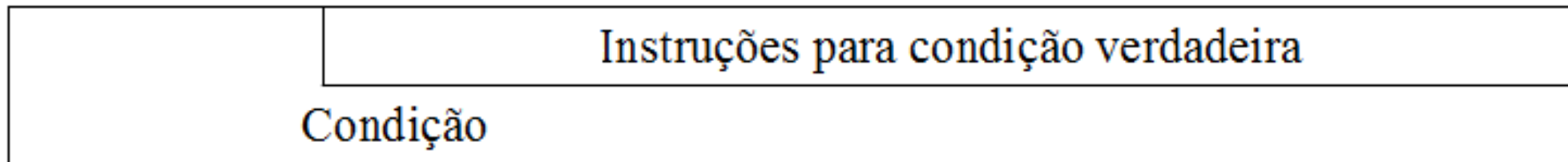
- Pseudocódigo

repita

 <instruções para condição verdadeira>

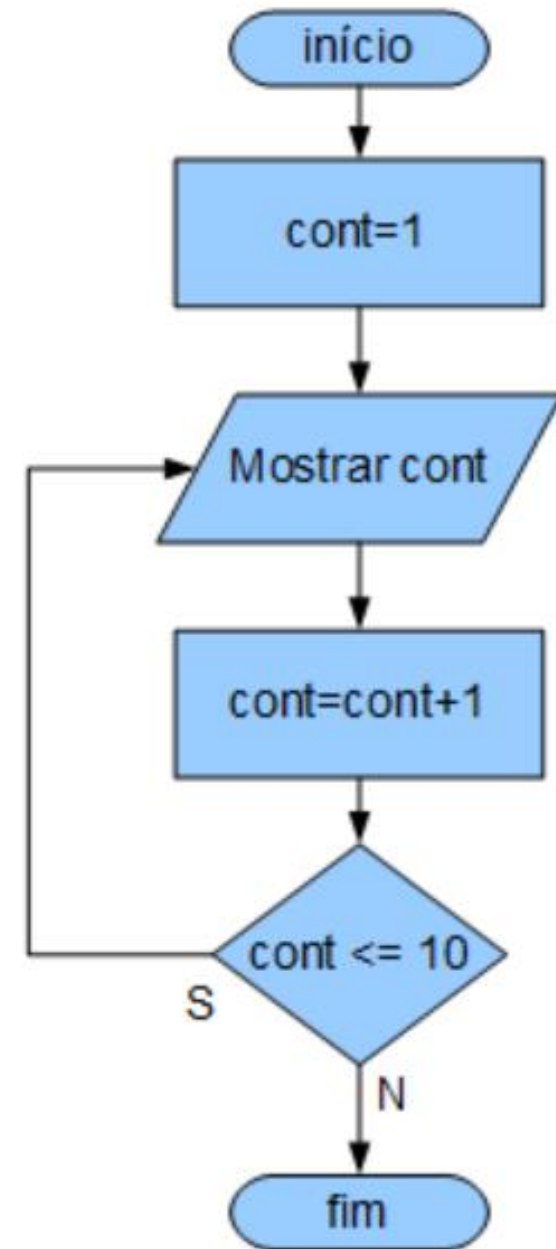
enquanto (<condição>)

- Diagrama de Chapin



Repita ... enquanto (do ... while)

- Mostrar os números de 1 a 10 na tela



Repetição repita enquanto em C

Comando `do...while`

```
do
{
    comandos;
} while (condição de teste);
```

- O laço `do-while` repete os comandos até que a condição se torne falsa.
- Ao contrário do laço `while`, ele verifica a condição no final do laço

- Exemplo:

```
do
{
    ...
    char = getchar();
}while (char != 'S' && char != 's' );
```

Repetição repita enquanto no Scilab

- O Scilab não apresenta uma estrutura **repita ... enquanto** dedicada.
- A estrutura **repita ... enquanto** pode ser implementada do seguinte modo

```
while 1 do
```

```
...
```

```
a=input('Sair <s> ou <n>', 's');
```

```
if a=='s' | a=='S' then //teste no fim
```

```
    break;
```

```
end
```

```
end
```



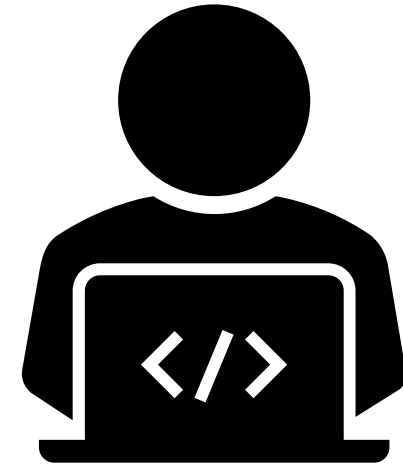
EXEMPLO 4

- FAZER UM PROGRAMA QUE PEÇA AO USUÁRIO UM NÚMERO INTEIRO E INFORME SE O NÚMERO É PAR OU ÍMPAR
- REPITA O PROCESSO ATÉ QUE O USUÁRIO DIGITE O NÚMERO ZERO

- Fazer um programa utilizando a estrutura **repita ... enquanto** que some os números que o usuário entrar até que ele digite o número 0.

Exemplo de uso:

```
//Em negrito, o que o
// usuário digita
Soma parcial é 0
Digite um número: 4
Soma parcial é 4
Digite um número: 2
Soma parcial é 6
Digite um número: 7
Soma parcial é 13
Digite um número: 0
Soma final é 13
Fim do programa
```



Exercício 4

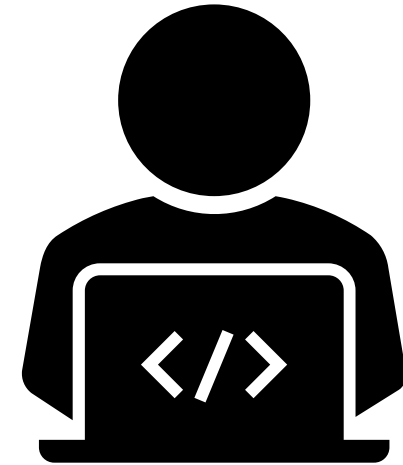
- Utilizando a estrutura **repita ... enquanto**, faça um programa que calcule a resistência em paralelo de um número n de resistores. Ao final de cada cálculo, o programa deve perguntar se o usuário deseja sair ou executar um novo cálculo.

Exercício 5

Exemplo de uso:

```
Informe o número de
resistores: 3
Entre com o resistor 1: 8
Entre com o resistor 2: 6
Entre com o resistor 3: 4
O Resultado é: 1.85 ohms
Deseja Sair? <S> ou <N>: N
```

```
Informe o número de
resistores: 4
Entre com o resistor 1: 8
Entre com o resistor 2: 16
Entre com o resistor 3: 24
Entre com o resistor 3: 12
O Resultado é: 3.20 ohms
Deseja Sair? <S> ou <N>: S
*** Programa encerrado ***
```



Bibliografia e crédito das figuras



OUALLINE, S. Practical C Programming. 3a ed. O'Reilly, 1997.



SEBESTA, R. Conceitos de Linguagens de Programação. 5a ed. Porto Alegre: Bookman, 2003.



FORBELLONE, A. L. V.; Eberspacher, H. F. Lógica de Programação – A construção de Algoritmos e Estrutura de Dados. 2ª. Ed. São Paulo: Pearson/Makron Books, 2000.



ASHLEY, Stephen. The Fundamentals of C. 1a ed. Kindle Edition.



http://help.scilab.org/docs/6.1.0/pt_BR/index.html



Compiler, assembler, linker and loader: a brief story. Disponível em: <http://www.tenouk.com/ModuleW.html>



<http://www.programmingbasics.org>