

PRIMEIROS EXERCICIOS DE PROGRAMAÇÃO

Uso (opcional) do Scilab

scilab-5.0.3

Consortium Scilab (DIGITEO)
Copyright (c) 1989-2008 (INRIA)
Copyright (c) 1989-2007 (ENPC)

<http://www.scilab.org>

Startup execution: loading initial environment

-->3.24^3-31.48+123.6/1.78
ans = 71.970426

-->x=2
x = 2.

-->who_user
User variables are:
x ans home
Using 26 elements out of 4965209

-->pwd
ans = C:\Documents and Settings\MFabbri\Desktop

-->cd d:\fabbri
ans = d:\fabbri

-->pwd
ans = d:\fabbri

-->x
x = 2.

-->x = x + 1
x = 3.

-->x = x/2
x = 1.5

-->x = x*sqrt(x)
x = 1.8371173

-->x = 1.5 + x*x + ...
-->x^3 + exp(x)
x = 17.353684

-->x = x^(1/3);

-->x
x = 1.3731254

-->y = sin(0.73); x = x/y;

-->x
x = 2.0590612

-->y
y = 0.6668696

-->%pi
%pi = 3.1415927

-->X
!--error 4
Undefined variable: X

```

-->format(20)

-->x
x = 2.05906115361923003

-->RaioDaBase = 2.3
RaioDaBase = 2.2999999999999998

-->Altura = 7.5
Altura = 7.5

-->Volume = %pi*Altura*RaioDaBase^2
Volume = 124.642688531175011

-->// vamos trocar os valores entre x e y

-->temp = x
temp = 2.05906115361923003

-->x = y
x = 0.66686963500369789

-->y = temp
y = 2.05906115361923003

```

TABELA DE CONFERENCIA

	x	y	temp
	2,06	0,667	?
temp = x	2,06	0,667	2,06
x = y	0,667	0,667	2,06
y = temp	0,667	2,06	2,06

Calcular $z = \frac{3,2 + \sqrt{5,8^2 + 2,13^2}}{0,68}$

```

-->// modo "desastrado"

-->z = (3.2+sqrt(5.8^2+2.13^2))/0.68
z = 13.7922732756342992

-->// melhor modo

-->z = 5.8^2 + 2.13^2
z = 38.1769000000000034

-->z = 3.2 + sqrt(z)
z = 9.37874582743132379

-->z = z/0.68
z = 13.7922732756342992

```

Vamos criar uma função que calcula a hipotenusa c de um triângulo retângulo de catetos a e b .

1. crie um arquivo de texto contendo o código abaixo:

```

function [c] = hyp(a,b)
    c = sqrt(a*a+b*b);
endfunction

```

2. Salve esse arquivo, por exemplo, com o nome “funcoes.txt”

```
-->getf('funcoes.txt')

-->who_user
User variables are:
hyp z temp Volume Altura RaioDaBase y cd x home
Using 539 elements out of 4965183

-->hyp(3,4)
ans = 5.
-->c = hyp(5.2 , 6.7)
c = 8.4811555816409836
-->x = hyp(hyp(3,5), 6)
x = 8.36660026534075563
```

Agora, uma função que toma o maior de dois números a e b.

1. Acrescente no final do arquivo “funcoes.txt” o código abaixo:

```
function [f] = Maior(a,b)
    if (a>=b) then f = a;
    else f = b;
    end
endfunction
```

2. Salve o arquivo.

```
-->getf('funcoes.txt')
Warning : redefining function: hyp . Use funcprot(0) to avoid this message

-->who_user
User variables are:
Maior help c hyp z temp Volume Altura RaioDaBase y cd x home
Using 1100 elements out of 4965174

-->Maior(3,5)
ans = 5.

-->Maior(3,3)
ans = 3.

-->Maior(3,-1)
ans = 3.

-->// O sinal '=' indica atribuição, enquanto que '==' indica comparação
-->X=2
X = 2.
-->X==2
ans = T
-->X==3
ans = F

-->if x==2 then print(%io(2),'x is 2'); else print(%io(2),'x is not 2'); end
x is 2

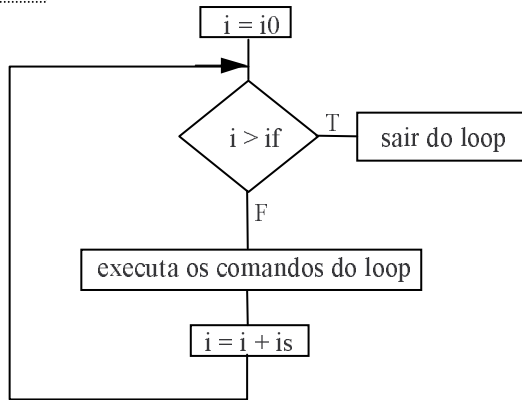
-->x=3
x = 3.

-->if x==2 then print(%io(2),'x is 2'); else print(%io(2),'x is not 2'); end
x is not 2
```

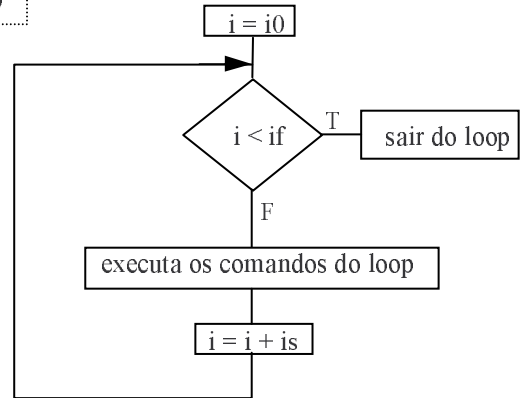
```
-->// soma dos dez primeiros numeros naturais
-->s = 0;
-->for i=1:10, s = s+i; end
-->s
s = 55.
```

Fluxograma do comando for i = i0:is:if, <comandos>

is > 0



is < 0



```
-->// soma dos dez primeiros numeros impares
-->si = 0;
-->for i=1:10, si = si+(2*i-1); end
-->si
si = 100.

-->// outro modo
-->si = 0;
-->for i=1:2:19, si = si+i; end

-->si
si = 100.

-->// calculando a fração (11+13+15+17+...+89)/(8+10+12+14+...+62)
-->num=11;den=8;...
--> for i=13:2:89, num=num+i; end...
--> for i=10:2:62, den=den+i; end...
--> f=num/den
f = 2.04081632653061229

-->// soma de todos os numeros naturais menores do que 35
-->s = 0; i = 1;
-->while i < 35, s = s+i; i = i+1; end
-->s
s = 595.

-->// outra forma
-->s=0; for i=1:34, s=s+i; end
-->s
s = 595.
```

```

-->// soma da serie harmônica 1+1/2+1/3+... até que o termo fique menor do que 0,01
-->s = 1; i = 1; t = 1;
-->while t>0.01, i = i+1; t = 1/i; s = s+t; end
-->s
s = 5.18737751763962063
-->i
i = 100.

-->// soma da serie 1-1/2+1/3-1/4+1/5-.... até que o termo fique menor do que 0,001
-->s = 1; i = 1; t = 1; sinal = 1;
-->while t>0.001, i = i+1; t = 1/i; sinal = -1*sinal; s = s+sinal*t; end
-->s
s = 0.69264743055982225

-->// a matemática nos informa que essa serie converge para ln(2)
-->log(2)
ans = 0.69314718055994529

--> // impressão formatada na tela (usando um comando do C)
-->for i=1:11, x=(i-1)*0.1; f=sqrt(x); fprintf(%io(2),'\nx=%f f=%f',x,f); end
x=0.000000 f=0.000000
x=0.100000 f=0.316228
x=0.200000 f=0.447214
x=0.300000 f=0.547723
x=0.400000 f=0.632456
x=0.500000 f=0.707107
x=0.600000 f=0.774597
x=0.700000 f=0.836660
x=0.800000 f=0.894427
x=0.900000 f=0.948683
x=1.000000 f=1.000000

```

EXERCÍCIOS

1. Escreva uma seqüência conveniente de passos computacionais para calcular

$$(a) \quad t = \frac{3,5^2 - \sqrt{2,7^2 + e^{1,4}}}{2,8 - 1,3^2}$$

$$(b) \quad r = \frac{e^{0,89} + 2e^{-1}}{3,21\sqrt{1,6^2 - 1,3}}$$

(O resultado das contas são (a) 8.00156 (b) 0.8800162)

2. Escreva na notação comum a quantidade que é calculada com a seqüência de passos abaixo:

$$(a) \quad \begin{aligned} s &= 2,37^2 + 8,79 \\ s &= s/6 \\ s &= s/\exp(-2,73) \end{aligned}$$

$$(b) \quad \begin{aligned} t &= -1,0*x*x/2 \\ t &= \exp(t) \\ t &= t/8 \\ t &= t/\sin(4*\pi*x) \\ t &= t*\tan(2*\pi*x) \end{aligned}$$

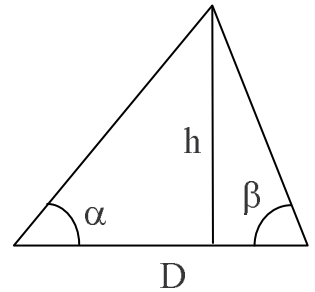
$$(c) \quad \begin{aligned} u &= 2 + 3,6^2 \\ u &= \sqrt{u} \\ u &= u/(1+u) \end{aligned}$$

$$Resp.: (a) \quad s = \frac{2,37^2 + 8,79}{6e^{-2,73}}$$

$$(b) \quad t = \frac{e^{-x^2/2}}{8\text{sen}(4\pi x)} \tan(2\pi x)$$

$$(c) \quad u = \frac{\sqrt{2+3,6^2}}{1+\sqrt{2+3,6^2}}$$

3. Programe uma função `Altura(Alfa, Beta, D)` que calcule a altura do triângulo de base `D` e ângulos de base `Alfa` e `Beta`. Os ângulos são fornecidos em graus (lembre-se de que as funções trigonométricas do sistema trabalham em radianos). Use a fórmula conhecida $h = \frac{D}{\cot(\alpha) + \cot(\beta)}$. O sistema provê a função tangente, `tan()`.



Alguns valores: `Altura(45,90,10)=10`. `Altura(45,45,10)=5`. `Altura(45,80,10)=8.501`

`Altura(85,80,10)= 37.905258`

4. Faça um fluxograma do comando `while`.

5. Use o comando `for` para calcular:

- a soma dos numeros naturais maiores do que 35 e menores do que 57 *(resp.: 966)*
- a soma dos numeros pares maiores do que 35 e menores do que 57 *(resp.: 506)*
- a soma dos numeros impares maiores do que 35 e menores do que 57 *(resp.: 460)*

6. Repita o exercício anterior usando o comando `while`.

7. A função `modulo(a,b)` retorna o resto da divisão entre os inteiros `a` e `b`. Use os comandos `for` e `if` para calcular a soma dos números divisíveis por sete que estão entre 52 e 149 (inclusive).

(resp.: 1421)

8. Calcular a fração $(1 \times 3 \times 5 \times 7 \times 9 \times \dots \times 23) / (2 \times 4 \times 6 \times 8 \times \dots \times 24)$

(resp.: 0.08059013)

9. Use o comando `for` para programar uma função `fat(N)` que retorna o fatorial do número `N`. Essa função deve:

- testar a entrada, retornando uma mensagem de erro se `N` for negativo;
- testar a entrada, retornando uma mensagem de erro se `N` for maior do que 15;
- se a entrada não for inteira, ela deve ser truncada usando a função `int()` do sistema;
- lembre-se de que $0! = 1$

Teste seu programa usando a função `gamma(x)` do sistema. Para `x` natural, $\text{gamma}(x) = (x-1)!$

10. Programe funções para calcular os `N` primeiros termos das séries abaixo:

(a) $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$

essa série converge para $\pi/4$

(b) $1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$

essa série converge para $\pi^2/6$

(c) $1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \frac{1}{13} - \dots$

essa série converge para $\frac{\pi\sqrt{3}}{9} + \frac{1}{3}\ln 2$

11. Modifique as funções do exercício anterior para que elas somem as séries até que o termo fique menor do que uma tolerância dada.

12. Imprima na tela uma tabela da função $f(x) = x - \tan(x)$ para `x` de 3,2 a 4,7 em intervalos de 0,1.