

PROGRAMAÇÃO DE COMPUTADORES I

SÉTIMA PARTE

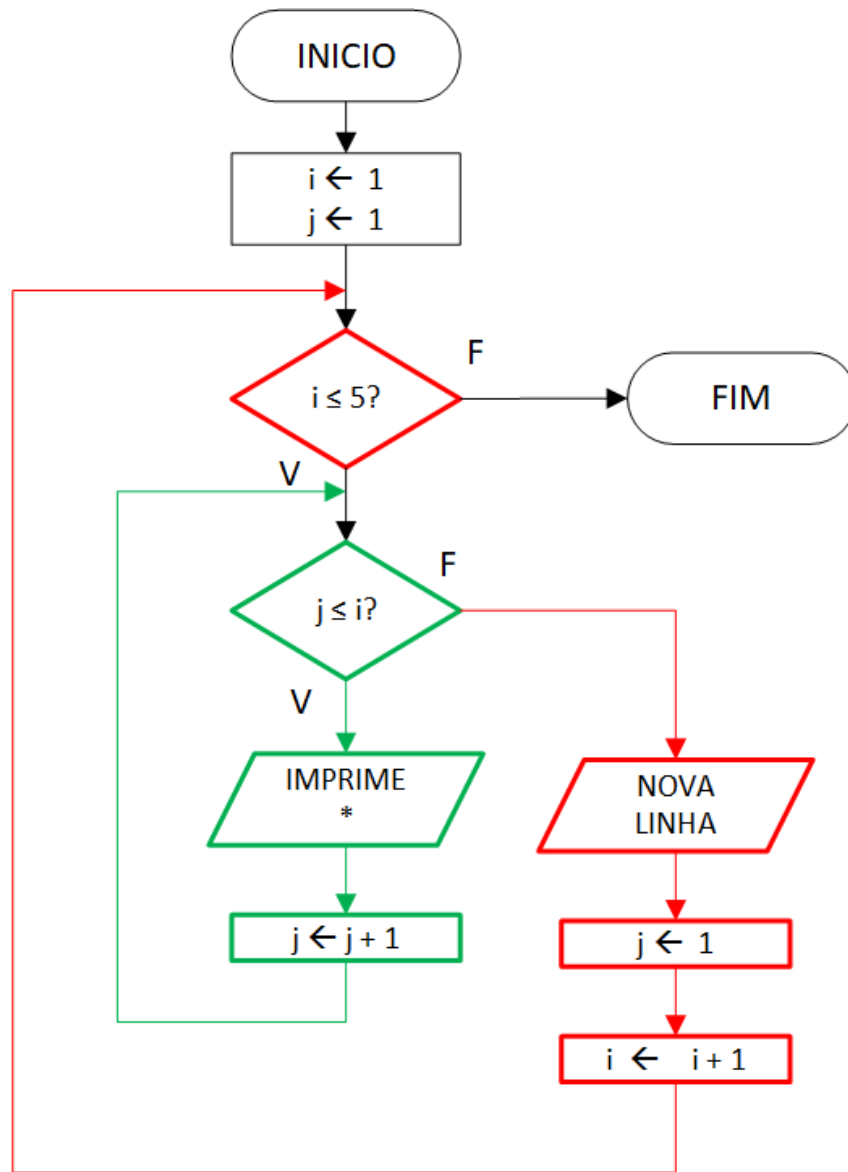
CAPÍTULO III
ESTRUTURAS DE
CONTROLE DE FLUXO

PARTE 2:
POR REPETIÇÃO:

- LAÇOS ANINHADOS
- PARA

Programação de Computadores I

prof. Marco Villaça



Laços aninhados

- É perfeitamente válido colocar um laço dentro de outro
- Em fluxogramas, lembre-se de que as setas de retorno não podem ser cruzadas.

EXEMPLO 1

- Faça um programa que imprima na tela a sequência (usando laços)

*

* *

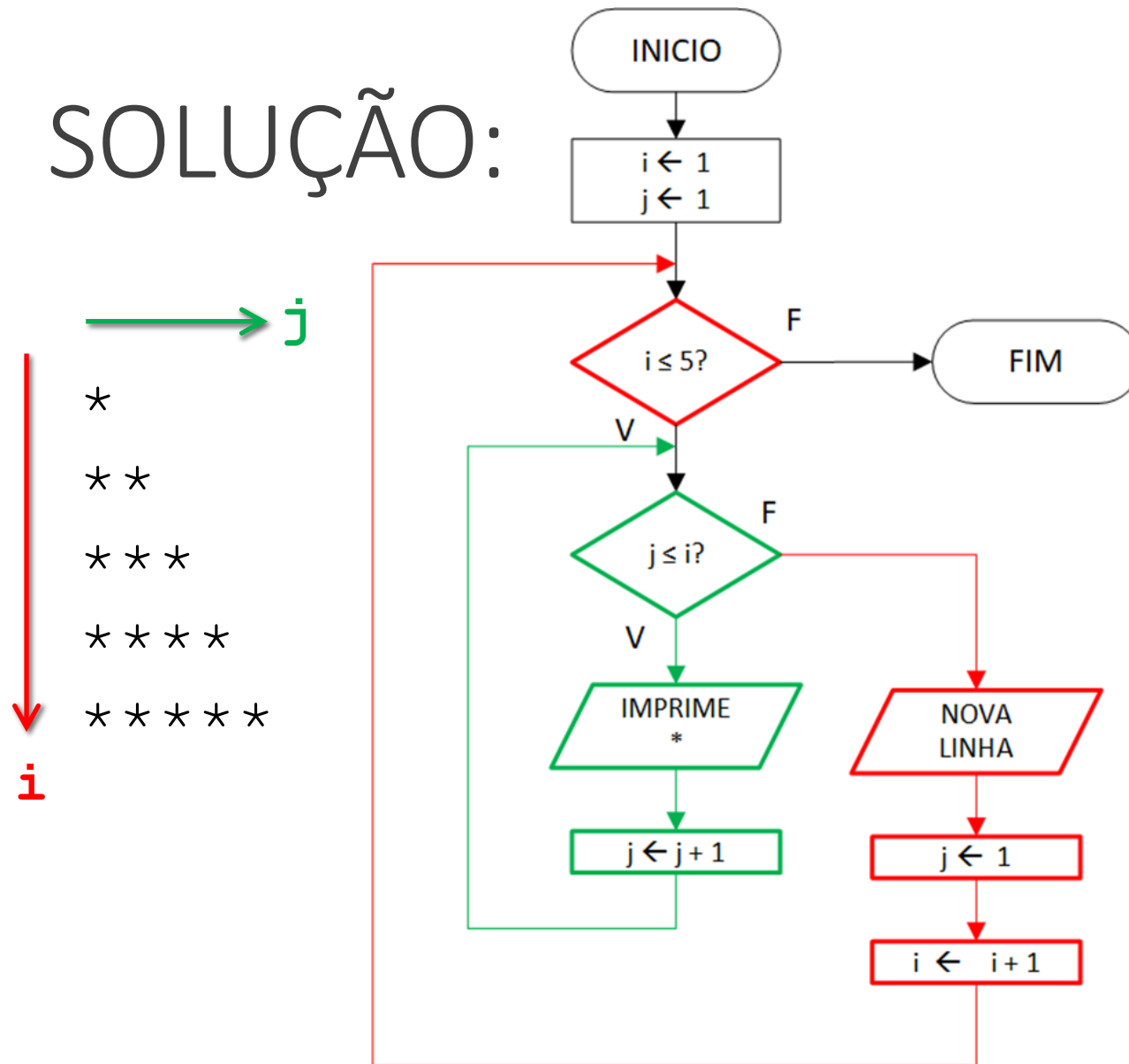
* * *

* * * *

* * * * *



SOLUÇÃO:



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i=1, j=1;
```

```
    while (i<=5)
```

```
    {
```

```
        while (j<=i)
```

```
        {
```

```
            printf("*");
```

```
            j++;
```

```
        }
```

```
        printf("\n");
```

```
        j = 1;
```

```
        i++;
```

```
    }
```

```
    return 0;
```

```
}
```

EXEMPLO 2

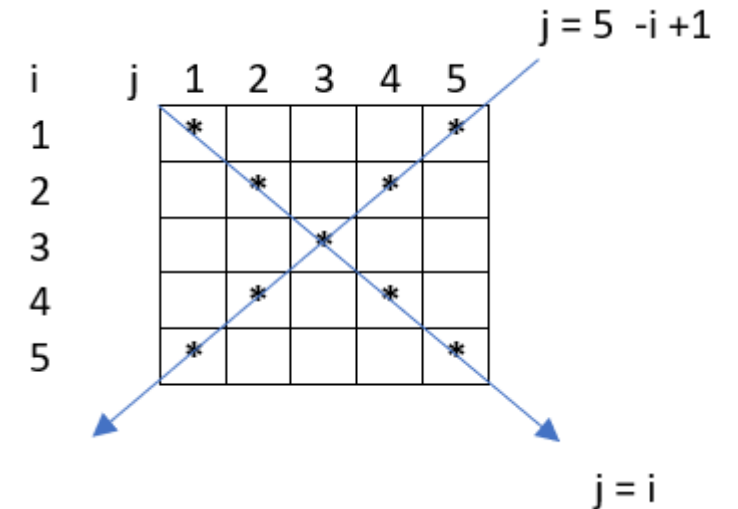
- Faça um programa que imprima na tela seguinte padrão(usando laços)

```
*      *  
      *  *  
     *  
    *  *  
   *      *
```



Exemplo 2 – construção da solução

- Nas casas onde $j = i$ ou onde $j = 5 - i + 1$ estão desenhados **asteriscos** e nas outras casas **nada** (espaço em branco).
- Assim, o programa conterà um **laço i** representando a posição da linha e **um laço interno j** representando a posição da coluna. Dentro do laço interno, **se $j = i$ ou $j = 5 - i + 1$** , será impresso um asterisco, **senão** um espaço em branco.
- Fora do laço interno, mas dentro do laço externo, o valor de j deve ser reinicializado e a impressão deve ser posicionada em uma nova linha.



EXEMPLO 3

- Faça um programa que imprima na tela seguinte padrão(usando laços)

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * * *  
* * *  
* *  
*
```



Exemplo 3 – construção da solução 1

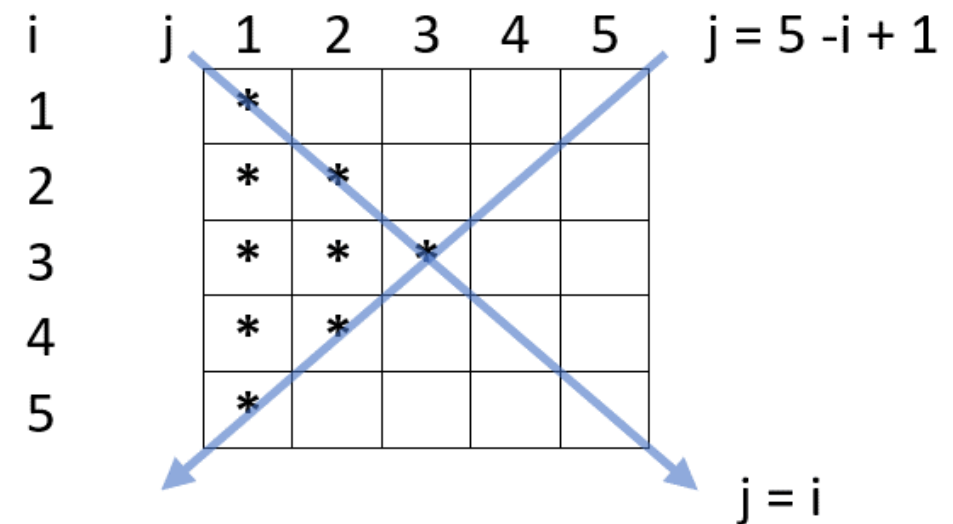
- Com i crescente, nas casas onde $j = i$ estão desenhados **asteriscos**.
- Após a linha central, i passa a ser decrescente e os asteriscos estão desenhados novamente onde $j = i$.
- Fora do laço interno, mas dentro do laço externo, o valor de j deve ser reinicializado e a impressão deve ser posicionada em uma nova linha.

i	j	1	2	3	4	5
1		*				
2		*	*			
3		*	*	*		
2		*	*			
1		*				

Simetria

Exemplo 3 – construção da solução 2

- Os asteriscos são impressos na área delimitada pelas retas $j = i$ e $j = 5 - i + 1$.
- Fora do laço interno, mas dentro do laço externo, o valor de j deve ser reinicializado e a impressão deve ser posicionada em uma nova linha.



Imprima na tela os seguintes padrões:

a.

1A2B3C4D5E

1A2B3C4D

1A2B3C

1A2B

1A

b. "Diamante"

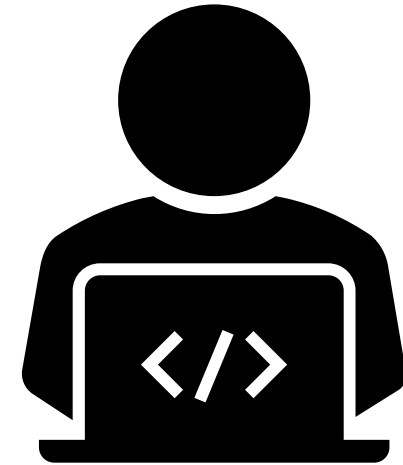
*

* * *

* * * * *

* * *

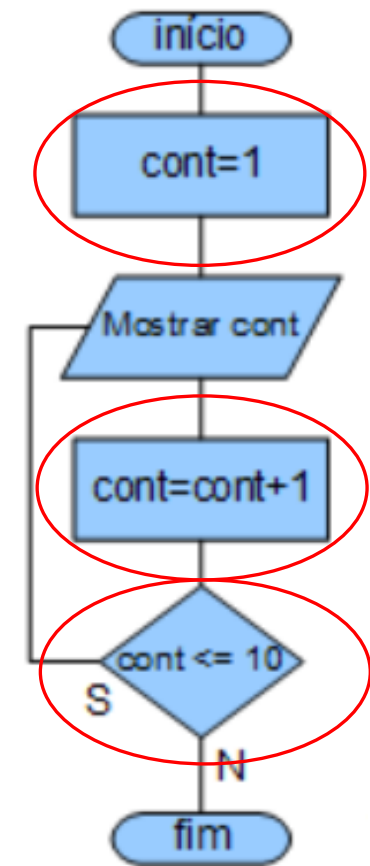
*



Exercício 1

Repetição para (for)

- Observe que nas duas estruturas (`while` e `do...while`), há três elementos
 - ✓ inicialização da variável de controle
 - ✓ fim (dado pela condição de teste)
 - ✓ passo ou atualização da variável

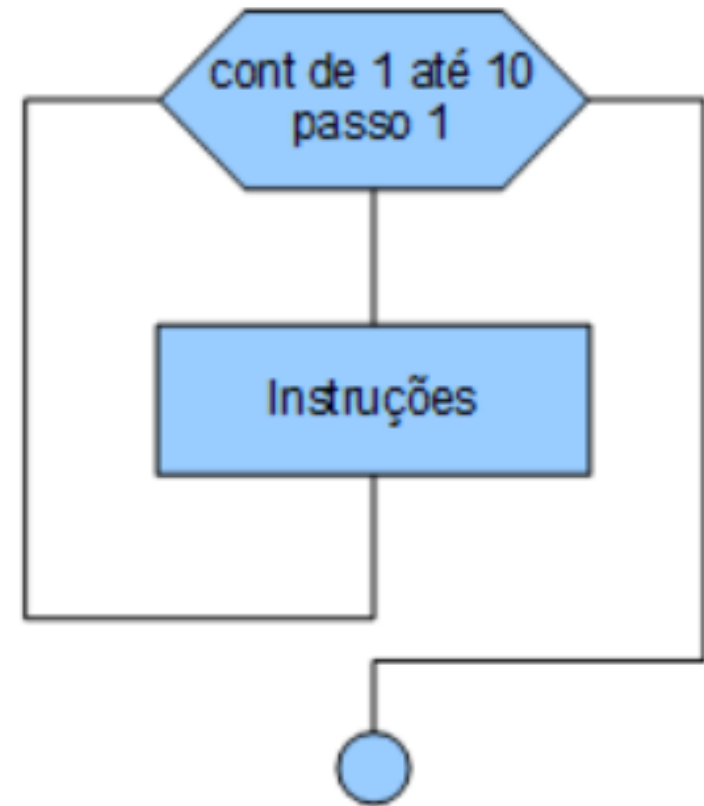


Repetição para (for)

- Nos exemplos anteriores, uma variável “contador” teve que ser inicializada previamente
- No laço com for, o contador faz parte da estrutura
- O número de ciclos (repetições) é definido no próprio comando

Repetição para (for)

- Símbolo
 - Hexágono
 - Indica preparação de uma variável com
 - Início
 - Fim
 - Passo



Laço para em C

Comando `for`

- Laço com comando `for`
- Executa um bloco de código enquanto uma condição pré-definida for verdadeira.

```
for (inicialização; condição_de_teste; atualização)
    comando;
```

- Equivale a

```
while (condição_de_teste) {
    comando;
    atualização;
}
```

Comando `for` versus `while`

```
/* imprimir números de  
0 até 9 */
```

```
/* Versão com WHILE */
```

```
int num=0;
```

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

```
/* imprimir números de  
0 até 9 */
```

```
/* Versão com FOR */
```

```
int num;
```

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


Laço para em C

Comando `for`

- No C99, a primeira expressão em um comando `for` pode ser substituída por uma declaração, permitindo ao programador declarar uma variável para uso no próprio laço.
- Esta variável não poderá ser acessada fora do corpo do laço

```
/* imprimir números de 0 até 9 */  
/* Versão do FOR para o C99*/  
for (int num = 0; num < 10; num++)  
{  
    printf("%d\n", num);  
}
```

Compilando no padrão C99

- No Code::Blocks para mudar para o padrão C99, ir para Settings → Compiler... → Compiler Flags e selecionar *“Have gcc follow the 1999 ISO C language standard”*.
- Em versões mais antigas ir para Settings → Compiler and debugger. In “other options” digitar “-std=c99”

Laço para no Scilab

Comando `for`

for <variavel> = <inicio>:[<passo>]:<fim>

- Se o <passo> for omitido, este será igual a 1.

Exemplo:

// imprime 1,3,5,7,9,11 na tela

```
for k =1:2:11
```

```
    disp(k)
```

```
end
```

EXEMPLO 4

- Chico tem 1,50 metros e cresce 1,8 centímetros por ano, enquanto Zé tem 1,41 metros e cresce 3,2 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Zé seja maior que Chico.



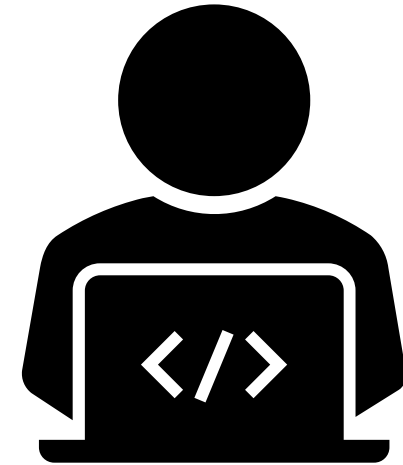
EXEMPLO 4

- Elaborar um programa em C que apresente o resultado inteiro da multiplicação de dois números positivos quaisquer.
- Para a elaboração do programa, não utilizar em hipótese alguma o operador aritmético da multiplicação.
- A solução deve ser alcançada com a utilização de laço. Ou seja, o programa deve realizar n somas sucessivas do multiplicando, onde n é o multiplicador



- Usando laço for, faça um programa que mostre na tela a sequência

150 50 16 5 1

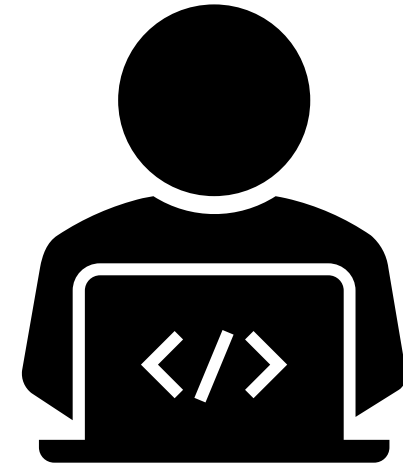


Exercício 2

Faça um programa que:

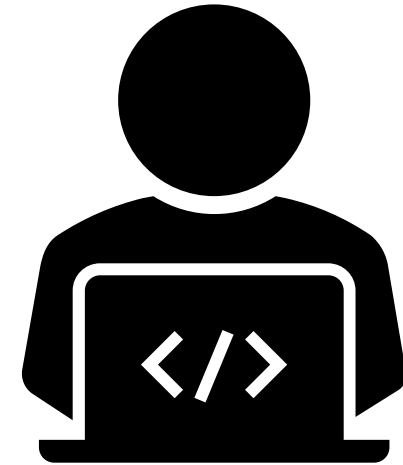
- Peça ao usuário 1 número n que seja potência de 2
- Verifique se o número é potência de 2. Caso não seja, solicite o número novamente
- Imprima na tela a sequência (usando laço for)

1 2 4 8 16 32 ... n .



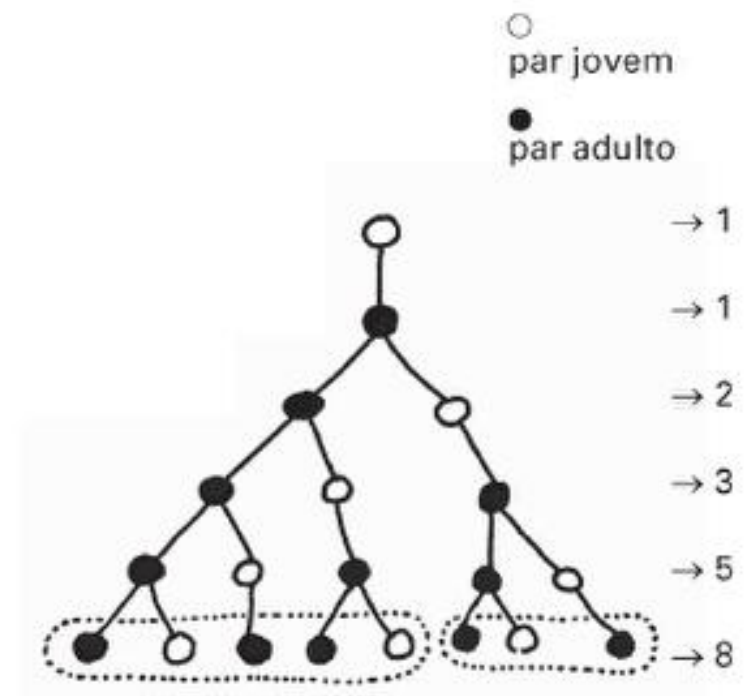
Exercício 3

- Elaborar um programa que apresente o resultado inteiro da divisão de dois números inteiros positivos quaisquer.
- Para a elaboração do programa, não utilizar em hipótese alguma o operador aritmético de divisão.
- A solução deve ser alcançada com a utilização de laço. Ou seja, o programa deve apresentar como resultado (quociente) quantas vezes o divisor cabe no dividendo.



Exercício 4

- Faça um programa que apresente a série de Fibonacci até o décimo quinto termo.
- A série de Fibonacci é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ...
- Essa sequência tem uma lei de formação simples:
 - Cada elemento, a partir do terceiro, é obtido somando-se os dois anteriores.
 - Veja: $1+1 = 2$, $2+1 = 3$, $3+2 = 5$...



Exercício 5



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>