

## Programas, Máquinas e Computação

1)

Traduza os programas monolíticos representados por fluxogramas em programas recursivos e iterativos.

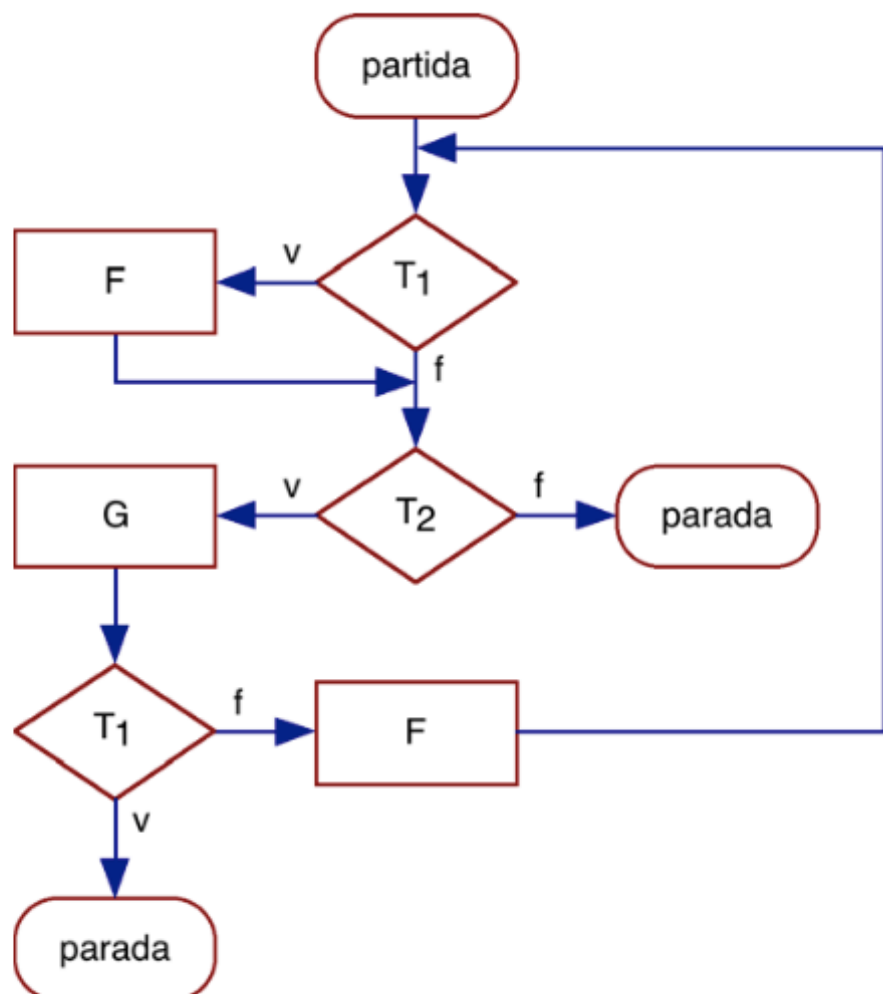
a) Fluxograma 1 representado na **Figura 2.24**;

b) Fluxograma 2 representado na **Figura 2.25**;

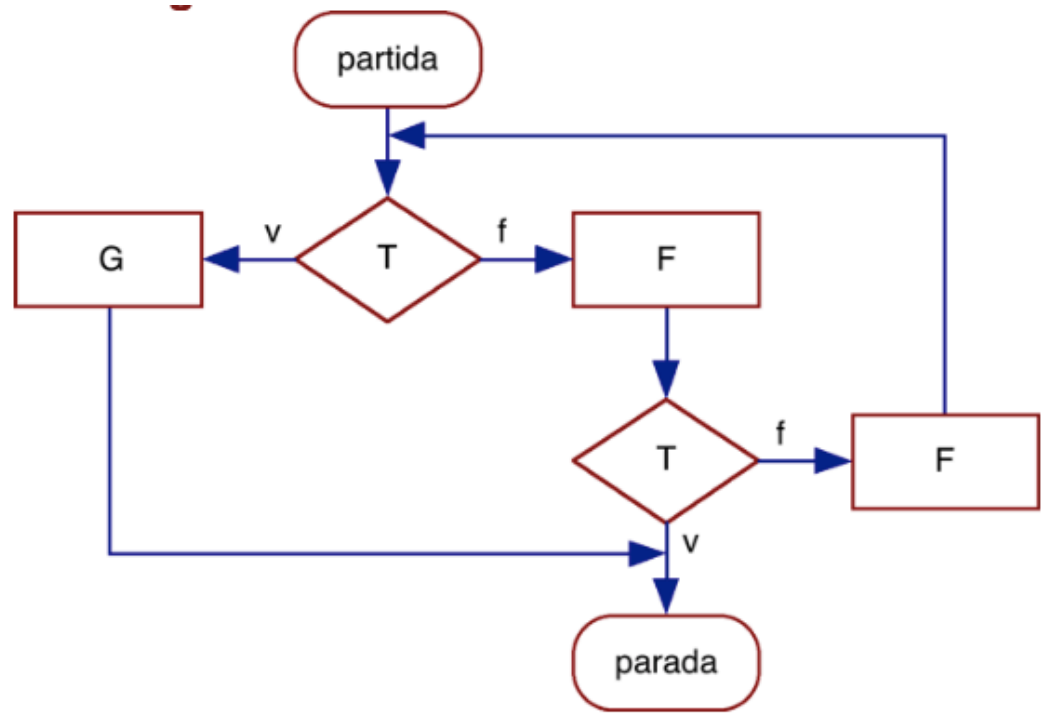
c) Fluxograma 3 representado na **Figura 2.26**;

d) Fluxograma 4 representado na **Figura 2.27**;

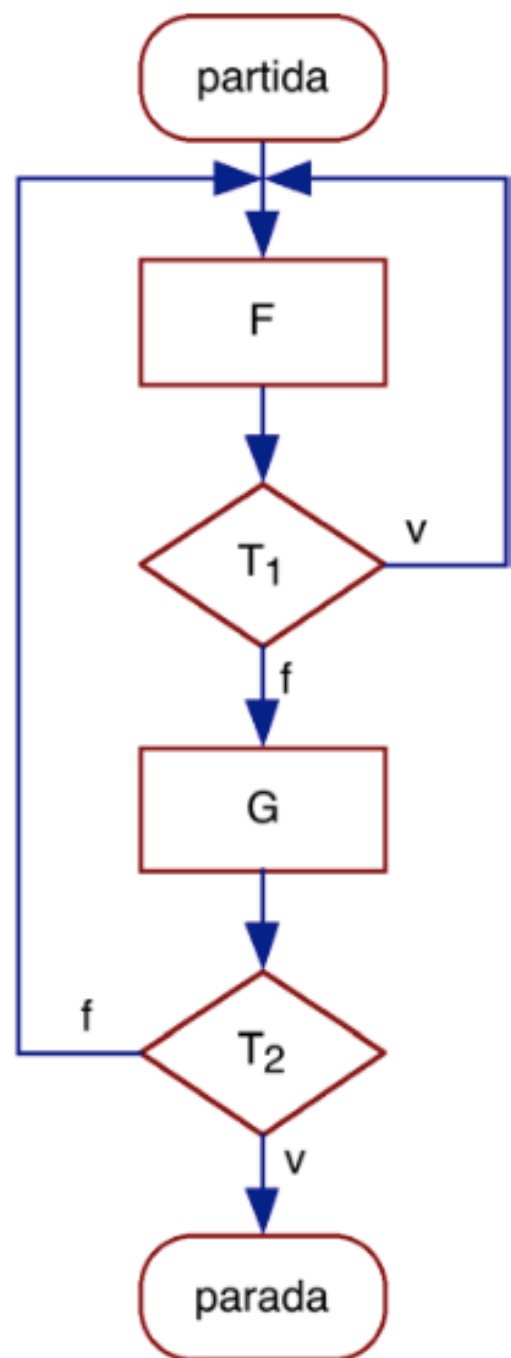
e) Fluxograma 5 representado na **Figura 2.28**.



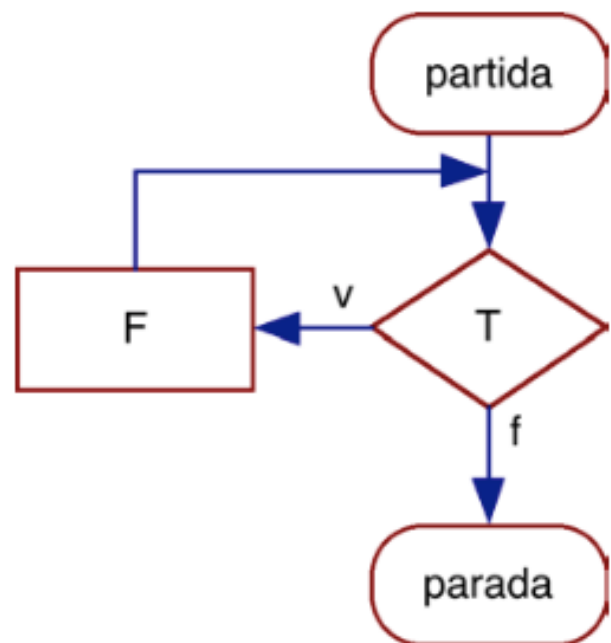
**Figura 2.24 Fluxograma 1**



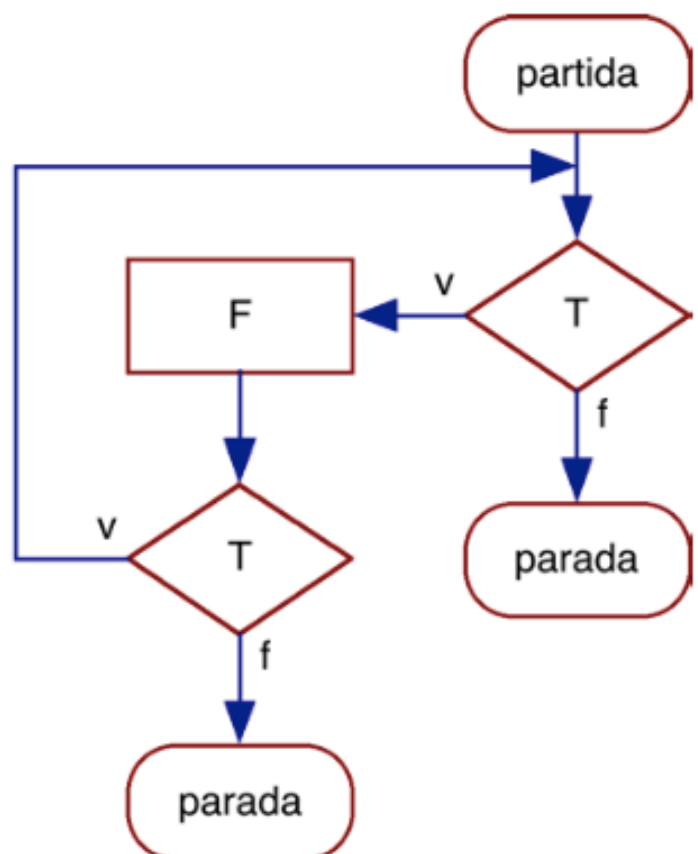
**Figura2.25 Fluxograma 2**



**Figura 2.26 Fluxograma 3**



**Figura 2.27 Fluxograma 4**



**Figura 2.28 Fluxograma 5**

2)

Traduza o programa iterativo representado na **Figura 2.29** em programa monolítico, nas formas de:

- a) Fluxograma;
- b) Instruções rotuladas.

```
(se T1
  então enquanto T2
    faça (até T3
      faça (V;W))
  senão (✓))
```

**Figura 2.29 Programa iterativo**

3)

Traduza o programa recursivo representado na **Figura 2.30** em programa iterativo.

```
P é R1 onde
  R1 def (se T então F;R2 senão R1),
  R2 def G; (se T então F;R1 senão ✓)
```

**Figura 2.30 Programa recursivo**

4)

Traduza os programas iterativos W<sub>1</sub> e W<sub>2</sub> definidos na Figura e na Figura para programas recursivos.

#### Programa Iterativo W<sub>1</sub>

```
enquanto T
  faça (F;(se T faça ✓ senão G))
```

**Figura 2.31 Programa iterativo W<sub>1</sub>**

#### Programa Iterativo W<sub>2</sub>

```
enquanto T
  faça (F;enquanto T faça (F);G)
```

**Figura 2.32 Programa iterativo W<sub>2</sub>**

5)

No contexto da equivalência forte, marque a alternativa correta:

- a) Para qualquer programa monolítico, existe um programa iterativo equivalente;
- b) Para qualquer programa recursivo, existe um programa monolítico equivalente;
- c) Para qualquer programa recursivo, existe um programa iterativo equivalente;
- d) Para qualquer programa monolítico, existe um programa recursivo equivalente;
- e) Nenhuma das alternativas anteriores está correta.

6)

Sobre Equivalência Forte de Programas, analise as afirmações abaixo:

- I. Dois programas são fortemente equivalentes se e somente se os dois são do mesmo tipo e suas funções computadas são iguais;
- II. Nem todo programa recursivo possui um monolítico fortemente equivalente pelo fato de que o primeiro é mais genérico que o segundo;

III. As funções computadas por programas fortemente equivalentes

possuem a propriedade de que as mesmas operações podem ser efetuadas em qualquer ordem, independente do significado das mesmas, pois a saída é a mesma.

Marque a alternativa correta:

- a) Apenas I;
- b) Apenas II;
- c) Apenas III;
- d) Apenas II e III;
- e) I, II e III

7)

Com relação à equivalência de programas, marque a alternativa que descreve corretamente a igualdade de funções parciais, tendo-se duas funções parciais

$f, g: X \rightarrow Y$  para cada  $x \in X$ :

a) Ambas as funções são definidas e  $f(x) = g(x)$ ;

b) Ou ambas são definidas e  $f(x)=g(x)$  ou ambas  $f(x)$  e  $g(x)$  são indefinidas;

c) Ou ambas são definidas e  $f(x)=g(x)$  ou pelo menos uma delas  $f(x)$  ou  $g(x)$  é

indefinida;

d) Funções parciais iguais são encontradas apenas em programas iterativos, pois

são os únicos que admitem funções;

e) Nenhuma das alternativas anteriores está correta.

## Programas Monolíticos com Fluxograma

8)

Desenvolver um programa monolítico, utilizando fluxograma, sobre uma máquina genérica, que calcule o fatorial de  $n$  utilizando a fórmula  $f = 1 * 2 * 3 * 4 * \dots * n$

O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo.

Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 5, o programa deverá apresentar como resposta o valor 120, ou seja,  $1 * 2 * 3 * 4 * 5$ . Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro

9)

Desenvolver um programa monolítico, utilizando fluxograma, sobre uma máquina genérica, que estime o valor da constante matemática e utilizando a fórmula

$$e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo.

Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 5, o programa deverá apresentar como resposta o valor 2.72, ou seja,  $1/0! + 1/1! + 1/2! + 1/3! + 1/4! + 1/5!$ .

Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

10)

Desenvolver um programa monolítico, utilizando fluxograma, sobre uma máquina genérica, que calcule o valor de  $e^x$  utilizando a fórmula  $e^x = x^0/0! + x^1/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo.

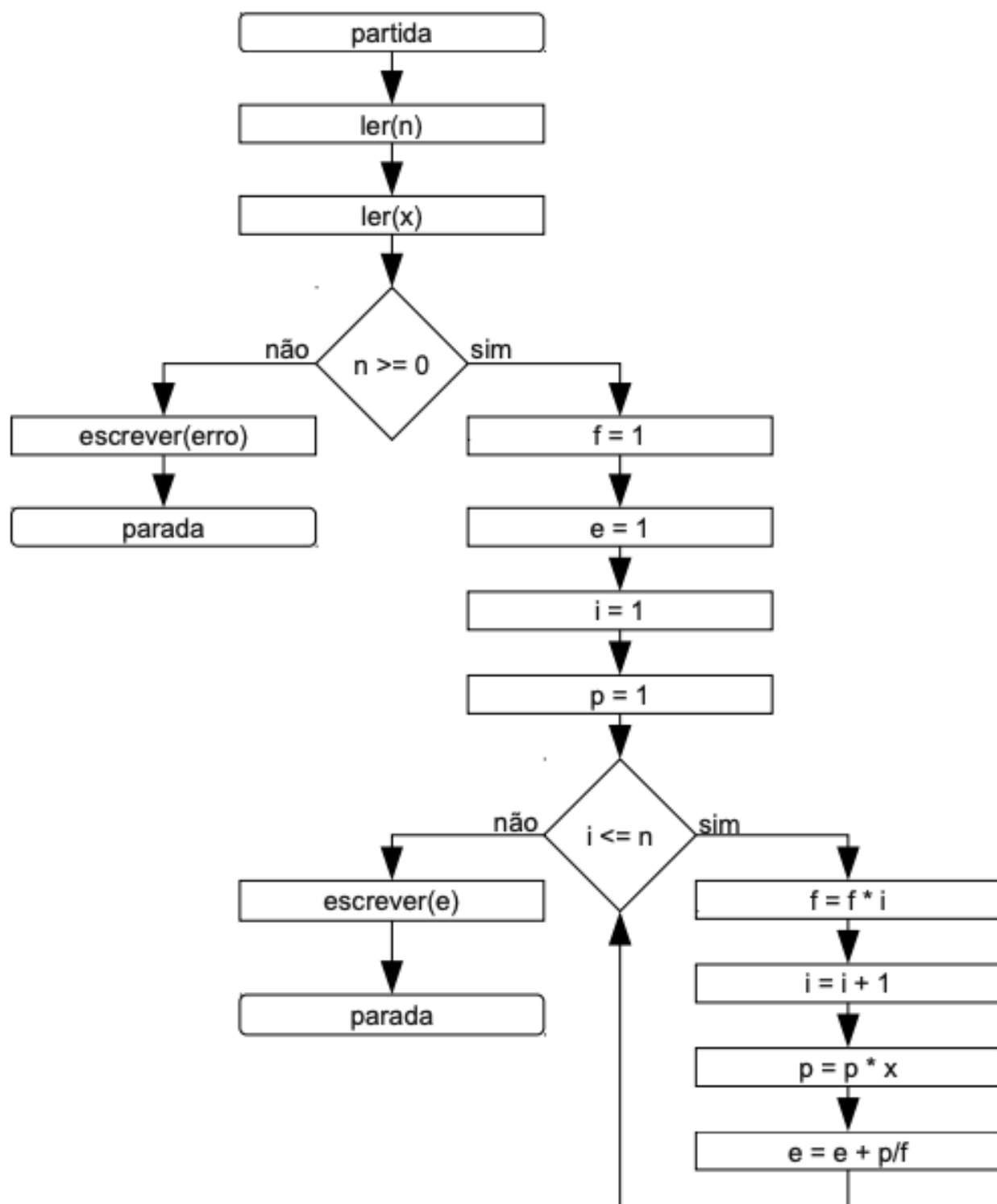
O valor de  $x$  será fornecido pelo usuário, podendo ser um valor (inteiro ou real) qualquer.

Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 4 e para  $x$  seja 2, o programa deverá apresentar como resposta o valor 7, ou seja,  $2^0/0! + 2^1/1! + 2^2/2! + 2^3/3! + 2^4/4!$ .

Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

**Resposta: veja como foi feito o fluxograma.**





11)

Desenvolver um programa monolítico, utilizando fluxograma, sobre uma máquina genérica, que apresente a série de Fibonacci. A série de Fibonacci é formada pela sequência 1, 1, 2, 3, 5, 8, 13, 21, 34, ... A série de Fibonacci é de grande importância matemática, e a lei básica é que a partir do terceiro termo, todos os termos são a soma dos dois últimos. O número de termos a serem impressos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos a serem impressos fornecido pelo usuário seja 7, o programa deverá apresentar como resposta a sequência de valores 1, 1, 2, 3, 5, 8, 13. Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

## Programas Monolíticos com Instruções Rotuladas

12)

Desenvolver um programa monolítico, utilizando instrução rotulada, sobre uma máquina genérica, que estime o valor da constante matemática  $e$  utilizando a fórmula  $e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ . O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 5, o programa deverá apresentar como resposta o valor 2.72, ou seja,  $1/0! + 1/1! + 1/2! + 1/3! + 1/4! + 1/5!$ . Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

**Resposta: preste atenção na solução**

R1 : Faça ler( $n$ ) vá\_para R2;  
R2 : Se  $n \leq 0$  então vá\_para R3 senão vá\_para R4;  
R3 : Faça escrever(erro) vá\_para Rx;  
R4 : Faça  $f = 1$  vá\_para R5;  
R5 : Faça  $e = 1$  vá\_para R6;  
R6 : Faça  $i = 1$  vá\_para R7;  
R7 : Se  $i \leq n$  então vá\_para R8 senão vá\_para R11;  
R8 : Faça  $f = f * i$  vá\_para R9;  
R9 : Faça  $i = i + 1$  vá\_para R10;  
R10: Faça  $e = e + 1/f$  vá\_para R7;  
R11: Faça escrever( $e$ ) vá\_para Rx;

13)

Desenvolver um programa monolítico, utilizando instrução rotulada, sobre uma máquina genérica, que calcule o valor de  $e^x$  utilizando a fórmula  $e^x = x^0/0! + x^1/1! + x^2/2! + x^3/3! + \dots + x^n/n!$ . O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo. O valor de  $x$  será fornecido pelo usuário, podendo ser um valor (inteiro ou real) qualquer. Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 4 e para  $x$  seja 2, o programa deverá apresentar como resposta o valor 7, ou seja,  $2^0/0! + 2^1/1! + 2^2/2! + 2^3/3! + 2^4/4!$ . Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

14)

Desenvolver um programa monolítico, utilizando instrução rotulada, sobre uma máquina genérica, que calcule o valor da série infinita  $S = 1 + 1/2 + 1/4 + 1/6 + 1/8 + 1/10 + \dots$ . O número de termos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos fornecido pelo usuário seja 5, o programa deverá apresentar como resposta o valor 2.04, ou seja,  $1 + 1/2 + 1/4 + 1/6 + 1/8$ . Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

15)

Desenvolver um programa monolítico, utilizando instrução rotulada, sobre uma máquina genérica, que apresente a série de Fibonacci. A série de Fibonacci é formada pela sequência 1, 1, 2, 3, 5, 8, 13, 21, 34, ... A série de Fibonacci é de grande importância matemática, e a lei básica é que a partir do terceiro termo, todos os termos são a soma dos dois últimos. O número de termos a serem impressos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos a serem impressos fornecido pelo usuário seja 7, o programa deverá apresentar como resposta a sequência de valores 1, 1, 2, 3, 5, 8, 13. Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

## Programas Iterativos

16)

Desenvolver um programa iterativo, sobre uma máquina genérica, que estime o valor da constante matemática  $e$  utilizando a fórmula  $e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ . O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 5, o programa deverá apresentar como resposta o valor 2.72, ou seja,  $1/0! + 1/1! + 1/2! + 1/3! + 1/4! + 1/5!$ . Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

**Resposta: Preste atenção na solução.**

```

programa
  ler(n);
  se (n >= 0) então
    fatorial = 1;
    e = 1;
    termo = 1;
    enquanto (termo <= n) faça
      fatorial = fatorial * termo;
      e = e + 1 / fatorial;
      termo = termo + 1;
    fim enquanto;
    escrever(e);
  senão
    escrever(erro);
  fim se;
fim programa.

```

17)

Desenvolver um programa iterativo, sobre uma máquina genérica, que calcule o valor da série infinita  $S = 1 + 1/2 + 1/4 + 1/6 + 1/8 + 1/10 + \dots$ . O número de termos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos fornecido pelo usuário seja 5, o programa deverá apresentar como resposta o valor 2.04, ou seja,  $1 + 1/2 + 1/4 + 1/6 + 1/8$ . Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

18)

Desenvolver um programa iterativo, sobre uma máquina genérica, que apresente a somatória dos termos da série de Fibonacci. A série de Fibonacci é formada pela sequência 1, 1, 2, 3, 5, 8, 13, 21, 34, ... A série de Fibonacci é de grande importância matemática, e a lei básica é que a partir do terceiro termo, todos os termos são a soma dos dois últimos. O número de termos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos fornecido pelo usuário seja 7, o programa deverá apresentar como resposta o valor 33, ou seja,  $1 + 1 + 2 + 3 + 5 + 8 + 13$ . Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

19)

Desenvolver um programa iterativo, sobre uma máquina genérica, que apresente a somatória de  $2k$ , sendo  $k = 1, 2, \dots, n$ . O número de termos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos fornecido pelo usuário seja 5, o programa deverá apresentar como resposta o valor 30, ou seja,  $2 * 1 + 2 * 2 + 2 * 3 + 2 * 4 + 2 * 5$ . Caso o usuário

forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

## Programas Recursivos

20)

Desenvolver um programa recursivo, sobre uma máquina genérica, que estime o valor da constante matemática  $e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ . O valor de  $n$  será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o valor fornecido pelo usuário para  $n$  seja 5, o programa deverá apresentar como resposta o valor 2.72, ou seja,  $1/0! + 1/1! + 1/2! + 1/3! + 1/4! + 1/5!$ . Caso o usuário forneça um valor inválido para  $n$ , o programa deverá apresentar uma mensagem de erro.

Reposta:

```
função fatorial(n)
  se (n > 0)
    então retornar n * fatorial(n - 1);
    senão retornar 1;
  fim se;
fim função;

função constante(n)
  se (n > 1)
    então retornar 1 / fatorial(n - 1) + constante(n - 1);
    senão retornar 1;
  fim se;
fim função;

programa
  ler(n);
  se(n > 0) então
    e = constante(n);
    escrever(e);
  senão
    escrever(erro);
  fim se;
fim programa;
```

20-b) Converter a solução acima para a notação formal de programas recursivos, conforme o exemplo abaixo:

### **Exemplo 2.6** **Programa Recursivo.**

$P$  é  $R;Z$  onde  
 $R \text{ def } F;(\text{se } T \text{ então } R \text{ senão } G;S),$   
 $S \text{ def } (\text{se } T \text{ então } \vee \text{ senão } F;R)$   
 $Z \text{ def } (\text{se } T \text{ então } G \text{ senão } F)$

**figura 2.6 - Programa recursivo**

21)

Desenvolver um programa recursivo, sobre uma máquina genérica, que calcule o valor da série infinita  $S = 1 + 1/2 + 1/4 + 1/6 + 1/8 + 1/10 + \dots$ . O número de termos será fornecido pelo usuário, devendo ser um valor inteiro e positivo. Por exemplo, caso o número de termos fornecido pelo usuário seja 5, o programa deverá apresentar como resposta o valor 2.04, ou seja,  $1 + 1/2 + 1/4 + 1/6 + 1/8$ . Caso o usuário forneça um valor inválido para o número de termos, o programa deverá apresentar uma mensagem de erro.

Fazer a solução usando a notação usual conforme a resposta do exercício 20 e conforme a notação formal dada pela figura acima.

## **Máquinas, Computações e Funções Computadas**

22)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A$ . Apresentar a computação e a função computada para as entradas 3 e 4.

**Resposta: Preste atenção na solução.**

R1: Se  $a\_zero$  então vá\_para Rx senão vá\_para R2;  
R2: Faça subtrair\_a vá\_para R3;  
R3: Faça adicionar\_b vá\_para R1;

Computação para o valor 3:

(R1, (3, 0))  
(R2, (3, 0))  
(R3, (2, 0))  
(R1, (2, 1))  
(R2, (2, 1))  
(R3, (1, 1))  
(R1, (1, 2))  
(R2, (1, 2))  
(R3, (0, 2))  
(R1, (0, 3))  
(Rx, (0, 3)) : 3 -> 3

23)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A + 2$ . Apresentar a computação e a função computada para as entradas 2 e 3.

24)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A - 2$ . Apresentar a computação e a função computada para as entradas 4 e 5.

25)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A * 3$ . Apresentar a computação e a função computada para as entradas 2 e 3.

26)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A / 3$ . Apresentar a computação e a função computada para as entradas 5 e 6.

27)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = (A / 2) * 3$ . Apresentar a computação e a função computada para as entradas 4 e 5.

28)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = A + (A \% 3)$ . Apresentar a computação e a função computada para as entradas 4 e 5.

29)

Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre a máquina 2\_REG, que implemente a função  $B = (A / 2) - 3$ . Apresentar a computação e a função computada para as entradas 8 e 11.

## Equivalências de Programas e Máquinas

---

30)

Desenvolver um programa iterativo e um programa recursivo, equivalente ao programa monolítico apresentado a seguir.

Programa Monolítico – Instruções Rotuladas

R1: Faça F vá\_para R2;

R2: Se T1 então vá\_para R1 senão vá\_para R3;

R3: Faça G vá\_para R4;

R4: Se T2 então vá\_para Rx senão vá\_para R1;

**Resposta: Preste atenção na solução**



## Programa Iterativo

```
programa
  F;
  enquanto T1 faça
    F;
  fim enquanto;
  G;
  enquanto não T2 faça
    F;
    enquanto T1 faça
      F;
    fim enquanto;
  G;
  fim enquanto;
fim programa.
```

## Programa Recursivo

```
função principal()
  loop1();
fim função;

função loop1()
  F;
  se T1 então loop1();
    senão loop2();
  fim se
fim função;

função loop2()
  G;
  se T2 então ;
    senão loop1();
  fim se
fim função;
```

31)

Desenvolver um programa iterativo e um programa recursivo, sobre uma máquina genérica, equivalentes ao programa monolítico apresentado a seguir.

Programa Monolítico – Instruções Rotuladas

R1: faça  $S = 0$  vá\_para R2;

R2: faça  $N = 1$  vá\_para R3;

R3: se  $N < 100$  então vá\_para R4 senão vá\_para R6;

R4: faça  $S = S + N$  vá\_para R5;

R5: faça  $N = N + 1$  vá\_para R3;

R6: faça escrever(S) vá\_para R7;