

01. Desenvolver um programa iterativo, sobre uma máquina genérica, equivalente ao programa monolítico apresentado a seguir.

Programa Monolítico – Instruções Rotuladas

```
R1: Faça A = 0 vá_para R2;  
R2: Faça B = 0 vá_para R3;  
R3: Faça C = 0 vá_para R4;  
R4: Faça ler(D) vá_para R5;  
R5: Se (B > D) então vá_para R6 senão vá_para R7;  
R6: Faça escrever(A) vá_para Rx;  
R7: Faça E = B + 1 vá_para R8;  
R8: Faça F = 1 vá_para R9;  
R9: Se (E > 0) então vá_para R10 senão vá_para R12;  
R10: Faça F = F * E vá_para R11;  
R11: Faça E = E - 1 vá_para R9;  
R12: Se (C = 0) então vá_para R13 senão vá_para R15;  
R13: Faça A = A + 1/F vá_para R14;  
R14: Faça C = 1 vá_para R17;  
R15: Faça A = A - 1/F vá_para R16;  
R16: Faça C = 0 vá_para R17;  
R17: Faça B = B + 1 vá_para R5;
```

02. Desenvolver um programa monolítico, utilizando instruções rotuladas, sobre uma máquina genérica, equivalente ao programa iterativo apresentado a seguir. Apresentar os valores impressos pelo programa caso o valor fornecido para a variável X seja 8.

Programa Iterativo

```
programa  
  A = 1;  
  B = 0;  
  S = 0;  
  ler(X);  
  enquanto (X > 0) faça  
    escrever(A);  
    S = S + A;  
    A = A + B;  
    B = A - B;  
    X = X - 1;  
  fim enquanto;  
  escrever(S);  
fim programa.
```

03. 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;
```

04. Identificar e comparar construções análogas às usadas nas definições de programas monolíticos, iterativo e recursivo em linguagens de programação como: Pascal, C, Java ou outra de seu conhecimento. Apresente exemplos para auxiliar na comparação entre as definições.

05. 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;

06. 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  $x = 1$  vá\_para R2;  
R2: faça  $total = 0$  vá\_para R3;  
R3: Se  $x \leq 10$  então vá\_para R4 senão vá\_para R8;  
R4: faça  $y = x * x$  vá\_para R5;  
R5: faça escrever(y) vá\_para R6;  
R6: faça  $total = total + y$  vá\_para R7;  
R7: faça  $x = x + 1$  vá\_para R3;  
R8: faça escrever(total) vá\_para R9;

07. Faça uma comparação entre programas monolíticos, iterativos e recursivos. Quais as semelhanças? Quais as diferenças? Exemplifique.

08. 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 R3 senão vá\_para R1;  
R3: Faça G vá\_para R4;  
R4: Se T2 então vá\_para R3 senão vá\_para R5;  
R5: Faça H vá\_para R6;  
R6: Se T3 então vá\_para R3 senão vá\_para Rx;

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

Programa Monolítico – Instruções Rotuladas

R1: Se T1 vá para R2 senão vá para R3;  
R2: Faça F1 vá para R4;  
R3: Faça F2 vá para R4;  
R4: Se T2 vá para R6 senão vá para R5;  
R5: Faça F3 vá para R1;

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

Programa Monolítico – Instruções Rotuladas

R1: Faça ler(n) vá\_para R2;  
R2: Se  $(n < 1)$  então vá\_para R3 senão vá\_para R4;  
R3: Faça escrever(erro) vá\_para Rx;  
R4: Faça  $y = 1$  vá\_para R5;  
R5: Se  $(n > 0)$  então vá\_para R6 senão vá\_para Rx;  
R6: Faça escrever(y) vá\_para R7;  
R7: Faça  $y = y + 2$  vá\_para R8;  
R8: Faça  $n = n - 1$  vá\_para R5;

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

Programa Monolítico – Instruções Rotuladas

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  $s = 0$  vá\_para R5;  
R5: Faça  $a = 0$  vá\_para R6;  
R6: Faça  $b = 1$  vá\_para R7;  
R7: Se  $(n > 0)$  então vá\_para R8 senão vá\_para R12;  
R8: Faça  $s = s + b$  vá\_para R9;  
R9: Faça  $b = b + a$  vá\_para R10;  
R10: Faça  $a = b - a$  vá\_para R11;  
R11: Faça  $n = n - 1$  vá\_para R7;  
R12: Faça escrever(s) vá\_para Rx;

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

Programa Monolítico – Instruções Rotuladas

R01: Faça ler(n) vá\_para R02;  
R02: Se  $n \leq 0$  então vá\_para R03 senão vá\_para R04;  
R03: Faça escrever(erro) vá\_para Rx;  
R04: Faça  $h = 0$  vá\_para R05;  
R05: Faça  $f = 1$  vá\_para R06;  
R06: Faça  $k = 0$  vá\_para R07;  
R07: Faça  $s = 1$  vá\_para R08;  
R08: Se  $n > k$  então vá\_para R09 senão vá\_para R18;  
R09: Faça  $k = k + 1$  vá\_para R10;  
R10: Faça  $p = 1$  vá\_para R11;  
R11: Faça  $j = k$  vá\_para R12;  
R12: Se  $j > 0$  então vá\_para R13 senão vá\_para R15;  
R13: Faça  $p = p * k$  vá\_para R14;  
R14: Faça  $j = j - 1$  vá\_para R12;  
R15: Faça  $f = f * k$  vá\_para R16;  
R16: Faça  $h = h + p / f * s$  vá\_para R17;  
R17: Faça  $s = s * (-1)$  vá\_para R08;  
R18: Faça escrever(h) vá\_para Rx;

13. Desenvolver um programa monolítico e um programa recursivo, equivalente ao programa recursivo apresentado a seguir, apresentado na linguagem de programação C.

```
#include <stdio.h>

int sub1(int a);
int sub2(int a);

int sub1(int a)
{
    printf("sub1 - %d\n", a);
    if (a > 0)
        return sub2(a - 1);
    else
        return 0;
}

int sub2(int a)
{
    printf("sub2 - %d\n", a);
    if (a > 0 && a % 2 == 0)
        return sub2(a - 1);
    else
        return sub1(a);
}

int main()
{
    printf("main - %d", sub1(10));
    return 0;
}
```

```
sub1 - 10
sub2 - 9
sub1 - 9
sub2 - 8
sub2 - 7
sub1 - 7
sub2 - 6
sub2 - 5
sub1 - 5
sub2 - 4
sub2 - 3
sub1 - 3
sub2 - 2
sub2 - 1
sub1 - 1
sub2 - 0
sub1 - 0
main - 0
```