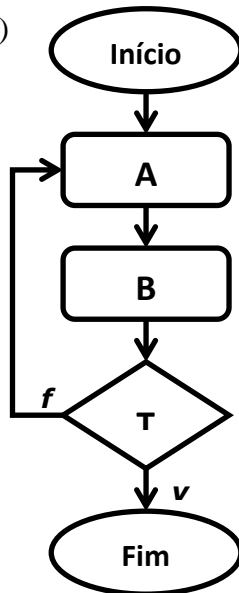


## ATIVIDADE EXTRA-CLASSE

### 1 – Programas, Máquinas e Computação GABARITO

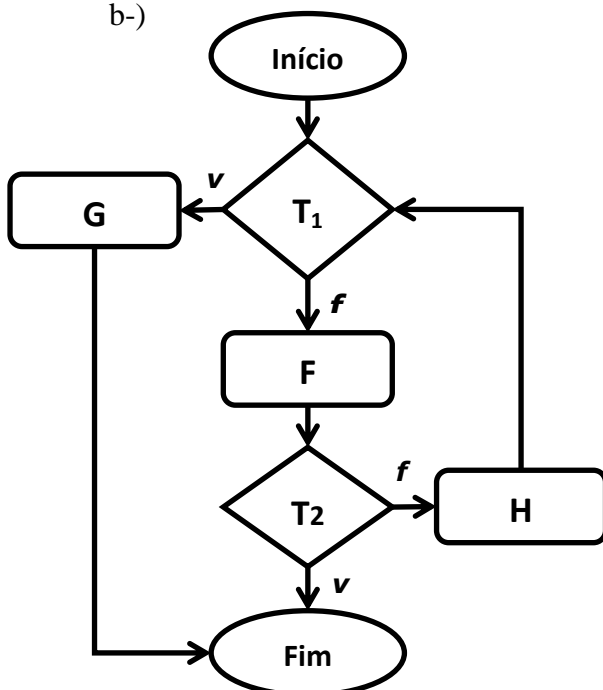
1-) Considere os programas monolíticos a seguir, descritos através de seus diagramas. Escreva seus algoritmos de forma descritiva.

a-)



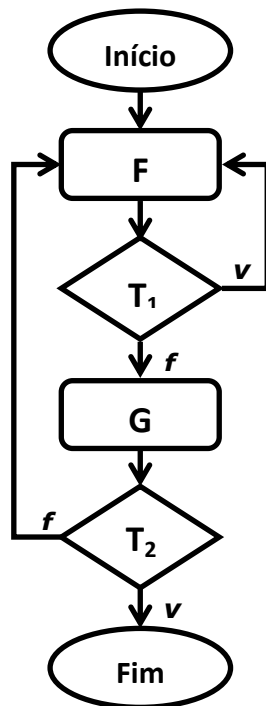
1: faça A; vá\_para 2  
2: faça B; vá\_para 3  
3: se T então vá\_para 4 senão vá\_para 1

b-)



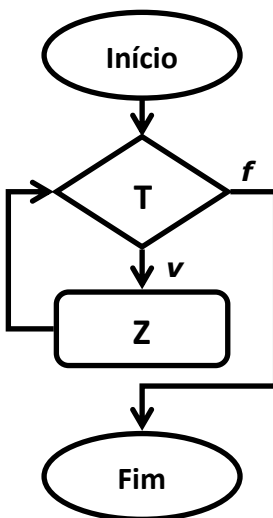
1: se T<sub>1</sub> então vá\_para 4 senão vá\_para 2  
2: faça F; vá\_para 3  
3: se T<sub>2</sub> então vá\_para 6 senão vá\_para 5  
4: faça G; vá\_para 6  
5: faça H; vá\_para 1

c-)



- 1: faça F; vá\_para 2
- 2: se  $T_1$  então vá\_para 1 senão vá\_para 3
- 3: faça G; vá\_para 4
- 4: se  $T_2$  então vá\_para 5 senão vá\_para 1

d-)



- 1: se T então vá\_para 2 senão vá\_para 3
- 2: faça Z; vá\_para 1

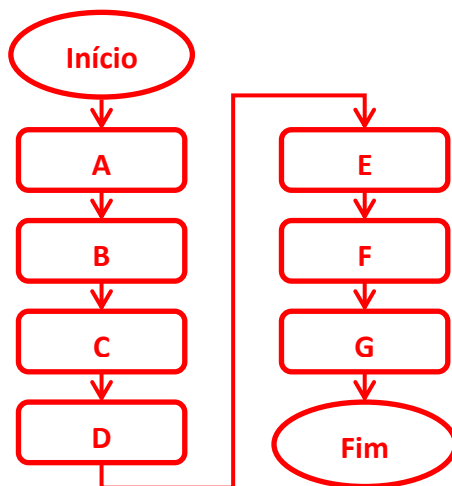
2-) Dado o algoritmo abaixo, converta-o para um Programa Monolítico.

```
var
nota1, nota2, media_ponderada: real
inicio
    escreva("Digite a nota 1: ")
    leia(nota1)
    escreva("Digite a nota 2: ")
    leia(nota2)
    media_ponderada <- (nota1 * 4 + nota2 * 6) / 10
    escreval("Notas: ", nota1, " e ", nota2)
    escreval("Média Ponderada = ", media_ponderada)
finalgoritmo
```

a-) Converta-o para um Programa Monolítico.

1: faça A; vá\_para 2  
 2: faça B; vá\_para 3  
 3: faça C; vá\_para 4  
 4: faça D; vá\_para 5  
 5: faça E; vá\_para 6  
 6: faça F; vá\_para 7  
 7: faça G; vá\_para 8

b-) Desenhe seu fluxograma na forma de diagrama.



c-) Converta-o para um Programa Iterativo.

I<sub>2</sub>=(A; B; C; D; E; F; G)

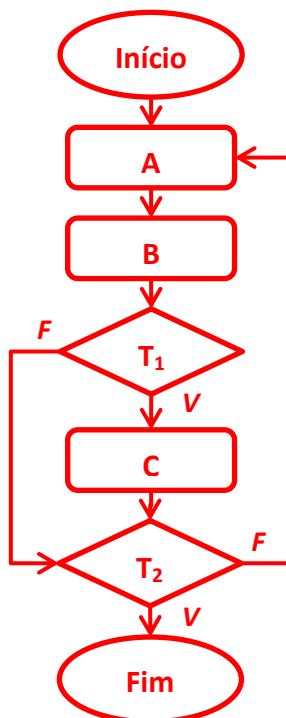
3-) Dado o trecho de código em Pascal a seguir.

```
var
sexo: string;
begin
  repeat
    write('Sexo (m/f): ');
    readln(sexo);
    if (sexo <> 'm') and (sexo <> 'f') then begin
      writeln('Sexo deve ser m ou f');
    end;
  until (sexo = 'm') or (sexo = 'f');
end.
```

a-) Converta-o para um Programa Monolítico.

- 1: faça A; vá\_para 2
- 2: faça B; vá\_para 3
- 3: se  $T_1$  então vá\_para 4 senão vá\_para 5
- 4: faça C; vá\_para 5
- 5: se  $T_2$  vá\_para 6 senão vá\_para 1

b-) Desenhe seu fluxograma na forma de diagrama.



c-) Converta-o para um Programa Iterativo.

$I_3 = (\text{até } T_1 \text{ faça } (A; B; \text{se } T_2 \text{ C senão } \checkmark))$

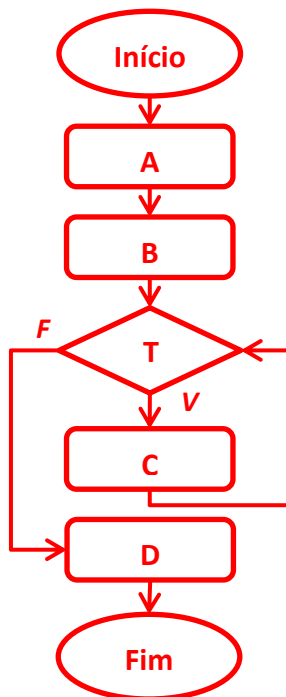
4-) Dado o trecho de código em Java abaixo.

```
public static void main(String[ ] args){
    int x = 9;
    int total = 1;
    for (int i = 1; i <= x; i++)
        total = total * i;
    System.out.println(x + "! = " + total);
}
```

a-) Converta-o para um Programa Monolítico.

**1: faça A; vá\_para 2**  
**2: faça B; vá\_para 3**  
**3: se T então vá\_para 4 senão vá\_para 5**  
**4: faça C; vá\_para 3**  
**5: faça D; vá\_para 6**

b-) Desenhe seu fluxograma na forma de diagrama.



c-) Converta-o para um Programa Iterativo.

**I<sub>4</sub> = (A; B; enquanto T faça (C); D)**

5-) Dado o trecho de código em Java a seguir, converta-o para um Programa Recursivo.

```
public static void main(String[ ] args){
    int x = 9;
    System.out.println(x + "! = " + fatorial(x));
}

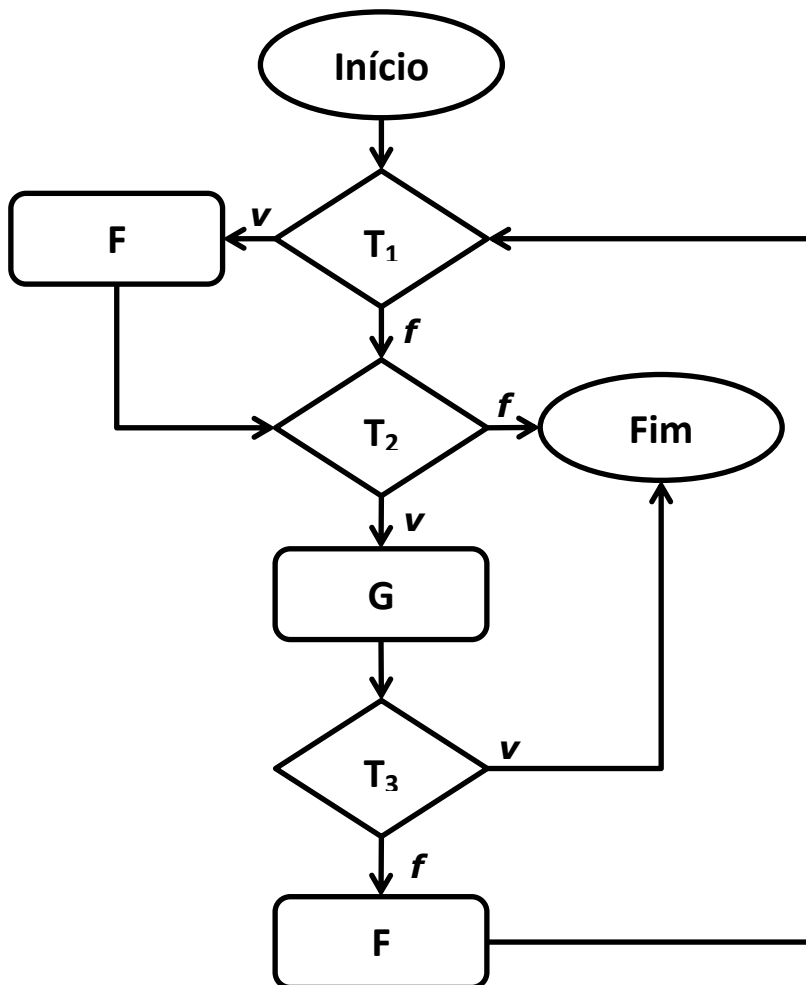
public static int fatorial(int x){
    if (x <= 1)
        return 1;
    else
        return x * fatorial(--x);
}
```

**P<sub>5</sub> é R, S onde**

**R def (A; S)**

**S def (se T então B senão S)**

6-) Dado o fluxograma de um programa monolítico a seguir, converta-o para um programa recursivo.



- 1: se T1 então vá\_para 2 senão vá para 3
- 2: faça F; vá\_para 3
- 3: se T2 então vá\_para 4 senão vá\_para 7
- 4: faça G; vá\_para 5
- 5: se T3 então vá\_para 7 senão vá\_para 6
- 6: faça F; vá\_para 1

$P_6$  é R, S onde

R def (se T<sub>1</sub> então (F; S) senão S)

S def (se T<sub>2</sub> então (G; (se T<sub>3</sub> então ✓ senão (F; R))) senão ✓)

7-) Considere a Máquina de 3 registradores  $M_{TRES}$  a seguir, descrita através de sua especificação.

$M_{TRES} = (\mathbb{N}^3, \mathbb{N}, \mathbb{N}, id_N, id_N, \{subtrai, acumula\}, \{ehZero\})$

- $\mathbb{N}^3$ : corresponde ao conjunto de valores de memória, e/s;
- $id_N: \mathbb{N}^3 \rightarrow \mathbb{N}^3$  é a função de entrada e saída;
- **acumula**:  $\mathbb{N}^3 \rightarrow \mathbb{N}^3$  é a interpretação tal que,  $\forall (n, m, o) \in \mathbb{N}^3$ ,  $acumula(n, m, o) = (n, m, o + m)$ ;
- **subtrai**:  $\mathbb{N}^3 \rightarrow \mathbb{N}^3$  é a interpretação tal que,  $\forall (n, m, o) \in \mathbb{N}^3$ ,  $subtrai(n, m, o) = (n - 1, m, o)$ ,  $n \neq 0$ ;  $subtrai(n, m) = 0$ , se  $n = 0$ ;
- **ehZero**:  $\mathbb{N}^3 \rightarrow \{\text{verdadeiro, falso}\}$  é a interpretação tal que,  $\forall (n, m, o) \in \mathbb{N}^3$ ,  $ehZero(n, m, o) = \text{verdadeiro}$ ,  $n = 0$ ;  $ehZero(n, m, o) = \text{falso}$ , se  $n \neq 0$ ;

E o programa monolítico do exercício 1a; sendo as operações A e B, respectivamente como as operações *acumula* e *subtrai*, definidas para a máquina de três registradores  $M_{TRES}$  e a operação de testes T, como a operação *ehZero* da mesma máquina.

Resolva a computação do programa monolítico para as respectivas entradas de valores a seguir:

**1: faça *acumula*; vá\_para 2**

**2: faça *subtrai*; vá\_para 3**

**3: se *ehZero* então vá\_para 4 senão vá\_para 1**

a-)  $v_0 = (4, 5, 0)$

(1, (4, 5, 0))	(3, (2, 5, 10))	(2, (1, 5, 20))
(2, (4, 5, 5))	(1, (2, 5, 10))	(3, (0, 5, 20))
(3, (3, 5, 5))	(2, (2, 5, 15))	(4, (0, 5, 20))
(1, (3, 5, 5))	(3, (1, 5, 15))	
(2, (3, 5, 10)) →	(1, (1, 5, 15)) →	

b-)  $v_1 = (3, 9, 0)$

(1, (3, 9, 0))	(2, (2, 9, 18))	(3, (0, 9, 27))
(2, (3, 9, 9))	(3, (1, 9, 18))	(4, (0, 9, 27))
(3, (2, 9, 9))	(1, (1, 9, 18))	
(1, (2, 9, 9)) →	(2, (1, 9, 27)) →	

c-)  $v_2 = (2, 7, 0)$

(1, (2, 7, 0))	(1, (1, 7, 7))	(4, (0, 7, 14))
(2, (2, 7, 7))	(2, (1, 7, 14))	
(3, (1, 7, 7)) →	(3, (0, 7, 14)) →	



8-) Considere a Máquina de 2 registradores  $M_{DOIS}$  a seguir, descrita através de sua especificação.

$M_{DOIS} = (\mathbb{N}^2, \mathbb{N}, \mathbb{N}, id_N, id_N, \{addX, addY, subX, subY, difXY\}, \{ehDif, ehIgual, ehMaior, ehMenor\})$

- $\mathbb{N}^2$ : corresponde ao conjunto de valores de memória, e/s;
- $id_N: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a função de entrada e saída;
- $addX: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $addX(x, y) = (x+1, y)$ ;
- $addY: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $addY(x, y) = (x, y+1)$ ;
- $subX: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $subX(x, y) = (x-1, y)$ ,  $x \neq 0$ ;  $subX(x, y) = (0, y)$ , se  $x=0$ ;
- $subY: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $subY(x, y) = (x, y-1)$ ,  $y \neq 0$ ;  $subY(x, y) = (x, 0)$ , se  $y=0$ ;
- $difXY: \mathbb{N}^2 \rightarrow \mathbb{N}^2$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $difXY(x, y) = (x-y, y)$ , se  $x > y$ ;  $difXY(x, y) = (x, y-x)$ , se  $y > x$ ;  $difXY(x, y) = (0, 0)$ , se  $x=y$ ;
- $ehDif: \mathbb{N}^2 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $ehDif(x, y) = \text{verdadeiro}$ , se  $x \neq y$ ;  $ehDif(x, y) = \text{falso}$ , se  $x=y$ ;
- $ehIgual: \mathbb{N}^2 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $ehIgual(x, y) = \text{verdadeiro}$ , se  $x=y$ ;  $ehIgual(x, y) = \text{falso}$ , se  $x \neq y$ ;
- $ehMaior: \mathbb{N}^2 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $ehMaior(x, y) = \text{verdadeiro}$ , se  $x > y$ ;  $ehMaior(x, y) = \text{falso}$ , se  $x \leq y$ ;
- $ehMenor: \mathbb{N}^2 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall (x, y) \in \mathbb{N}^2$ ,  $ehMenor(x, y) = \text{verdadeiro}$ , se  $x < y$ ;  $ehMenor(x, y) = \text{falso}$ , se  $x \geq y$ ;

E o programa iterativo  $I_{EX}$  a seguir.

```

IEX = (
  se ehIgual então (
    addX; addX; addY; addY) senão (
      se ehMaior então (até ehIgual (subX); addY; addY)
    senão (
      até ehIgual (addX); addX; addX ))
)

```

Resolva a computação do programa iterativo para as respectivas entradas de valores a seguir:

a-)  $v_0 = (3, 4)$

(se ehIgual então (

$addX; addX; addY; addY$ ) senão (

se ehMaior então (até ehIgual (subX); addY; addY) senão(

até ehIgual (addX); addX; addX )); ✓, (3,4))

(se ehMaior então (até ehIgual (subX); addY; addY) senão(

até ehIgual (addX); addX; addX; ✓, (3,4))

(até ehIgual (addX); addX; addX; ✓, (3,4))

(addX; até ehIgual (addX); addX; addX; ✓, (3,4))

(até ehIgual (addX); addX; addX; ✓, (4,4))

(addX; addX; ✓, (4,4))

(addX; ✓, (5,4))

(✓, (6,4))

b-)  $v_1=(5, 5)$

(se ehIgual então (

addX; addX; addY; addY) senão (

se ehMaior então (até ehIgual (subX); addY; addY) senão(

até ehIgual (addX); addX; addX )); ✓, (5,5))

(addX; addX; addY; addY; ✓, (5,5))

(addX; addY; addY; ✓, (6,5))

(addY; addY; ✓, (7,5))

(addY; ✓, (7,6))

(✓, (7,7))

c-)  $v_2=(8, 6)$

(se ehIgual então (

addX; addX; addY; addY) senão (

se ehMaior então (até ehIgual (subX); addY; addY) senão(

até ehIgual (addX); addX; addX )); ✓, (8,6))

(se ehMaior então (até ehIgual (subX); addY; addY) senão(

até ehIgual (addX); addX; addX; ✓, (8,6))

(até ehIgual (subX); addY; addY; ✓, (8,6))

(subX; até ehIgual (subX); addY; addY; ✓, (8,6))

(até ehIgual (subX); addY; addY; ✓, (7,6))

(subX; até ehIgual (subX); addY; addY; ✓, (7,6))

(até ehIgual (subX); addY; addY; ✓, (6,6))

(addY; addY; ✓, (6,6))

(addY; ✓, (6,7))

(✓, (6,8))

9-) Considerando a mesma máquina  $M_{DOIS}$  definida anteriormente, e o programa recursivo  $D_{REC}$  a seguir.

$D_{REC}$  é  $Q, R, S, T$  onde:  
 $Q$  def (se ehMaior então R senão (se ehMenor então S senão T))  
 $R$  def (addY; Q)  
 $S$  def (subX; Q)  
 $T$  def (addY; addX; ✓)

Resolva a computação do programa recursivo para as respectivas entradas de valores a seguir:

a-)  $v_0=(7,9)$

( $D_{REC}$ ; ✓, (7,9))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (7,9))

(se ehMenor então S senão T; ✓, (7,9))

(S; ✓, (7,9))

(subX; Q; ✓, (7,9))

(Q; ✓, (6,9))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (6,9))

(se ehMenor então S senão T; ✓, (6,9))

(S; ✓, (6,9))

(subX; Q; ✓, (6,9))

(Q; ✓, (5,9))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (5,9))

(se ehMenor então S senão T; ✓, (5,9))

(S; ✓, (5,9))

(subX; Q; ✓, (5,9))

(Q; ✓, (4,9))

...

*Obs: vai entrar em loop infinito.*

b-)  $v_1=(3,3)$

( $D_{REC}$ ; ✓, (3,3))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (3,3))

(se ehMenor então S senão T; ✓, (3,3))

(T; ✓, (3,3))

(addY; addX; ✓, (3,3))

(addX; ✓, (3,4))

(✓, (4,4))

c-)  $v_2=(4,1)$

(D<sub>REC</sub>; ✓, (4,1))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (4,1))

(R; ✓, (4,1))

(addY; Q; ✓, (4,1))

(Q; ✓, (4,2))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (4,2))

(R; ✓, (4,2))

(addY; Q; ✓, (4,2))

(Q; ✓, (4,3))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (4,3))

(R; ✓, (4,3))

(addY; Q; ✓, (4,3))

(Q; ✓, (4,4))

(se ehMaior então R senão (se ehMenor então S senão T); ✓, (4,4))

(se ehMenor então S senão T; ✓, (4,4))

(T; ✓, (4,4))

(addY; addX; ✓, (4,4))

(addX; ✓, (4,5))

(✓, (5,5))

10-) Dada a máquina  $M_{EX10}=(V, X, Y, \pi_X, \pi_Y, \Pi_O, \Pi_T)$ , definida a seguir:

$M_{EX10}=(\mathbb{Z}^3, \mathbb{Z}, \mathbb{Z}, id_N, id_N, \Pi_O, \Pi_T)$

- $\mathbb{Z}^3$ : corresponde ao conjunto de valores de memória, e/s;
- $id_N: \mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a função de entrada e saída.

a-) Defina a formalização das funções de operação  $\Pi_O$  e  $\Pi_T$ , baseado nos 3 registradores  $(a, b, c)$  conforme definido a seguir:

- $a = a + 1$   
**incA:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  $incA(a,b,c)=(a+1,b,c)$ ;**
- $b = b + 1$   
**incB:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  $incB(a,b,c)=(a,b+1,c)$ ;**
- $c = c + 1$   
**incC:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  $incC(a,b,c)=(a,b,c+1)$ ;**
- $a = a - 1$   
**decA:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  $decA(a,b,c)=(a-1,b,c)$ ;**
- $b = b - 1$   
**decB:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  $decB(a,b,c)=(a,b-1,c)$ ;**
- $c = c - 1$

- decC:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**decCB(a,b,c)=(a,b,c-1);****
- $c = a + b$   
**soma:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**soma(a,b,c)=(a,b,a+b);****
- $c = a - b$   
**subtrai:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**subtrai(a,b,c)=(a,b,a-b);****
- $c = a * b$   
**multiplica:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**multiplica(a,b,c)=(a,b,a\*b);****
- $c = a / b$   
**divide:  $\mathbb{Z}^3 \rightarrow \mathbb{Z}^3$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**divide(a,b,c)=(a,b,a/b);****
- $a = b$   
**igualAB:  $\mathbb{Z}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**igualAB(a,b,c)=veradeiro se  $a = b$ , igualAB(a,b,c)=falso se  $a \neq b$ ;****
- $b = c$   
**igualBC:  $\mathbb{Z}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**igualBC(a,b,c)=veradeiro se  $b = c$ , igualAB(a,b,c)=falso se  $b \neq c$ ;****
- $a = c$   
**igualAC:  $\mathbb{Z}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  $\forall(a, b, c) \in \mathbb{Z}^3$ ,  
**igualAC(a,b,c)=veradeiro se  $a = c$ , igualAB(a,b,c)=falso se  $a \neq c$ ;****
- $a = b = c$   
**igualABC:  $\mathbb{Z}^3 \rightarrow \{\text{verdadeiro}, \text{falso}\}$  é a interpretação tal que,  
 $\forall(a, b, c) \in \mathbb{Z}^3$ , **igualABC(a,b,c)=veradeiro se  $a = b$  e  $a = c$ ,  
igualAB(a,b,c)=falso se  $a \neq b$  ou  $a \neq c$ ;****

b-) Defina um programa iterativo **I<sub>EX10</sub>** que faça um loop com 10 iterações.

**I<sub>EX10</sub>=(até igualAB faça(incA; incC))**

c-) Compute este programa **I<sub>EX10</sub>** na máquina **M<sub>EX10</sub>**, com valores **v<sub>0</sub>** que atendam ao item b definido anteriormente.

**(até igualAB faça(incA; incC); ✓, (0, 10, -6))**  
**(incA; incC; até igualAB faça(incA; incC); ✓, (0, 10, -6))**  
**(incC; até igualAB faça(incA; incC); ✓, (1, 10, -6))**  
**(até igualAB faça(incA; incC); ✓, (1, 10, -5))**  
**(incA; incC; até igualAB faça(incA; incC); ✓, (1, 10, -5))**  
**(incC; até igualAB faça(incA; incC); ✓, (2, 10, -5))**  
**(até igualAB faça(incA; incC); ✓, (2, 10, -4))**  
**(incA; incC; até igualAB faça(incA; incC); ✓, (2, 10, -4))**  
**(incC; até igualAB faça(incA; incC); ✓, (3, 10, -4))**  
**(até igualAB faça(incA; incC); ✓, (3, 10, -3))**  
**(incA; incC; até igualAB faça(incA; incC); ✓, (3, 10, -3))**  
**(incC; até igualAB faça(incA; incC); ✓, (4, 10, -3))**  
**(até igualAB faça(incA; incC); ✓, (4, 10, -2))**

(incA; incC; até igualAB faça(incA; incC); ✓, (4, 10, -2))  
 (incC; até igualAB faça(incA; incC); ✓, (5, 10, -2))  
 (até igualAB faça(incA; incC); ✓, (5, 10, -1))  
 (incA; incC; até igualAB faça(incA; incC); ✓, (5, 10, -1))  
 (incC; até igualAB faça(incA; incC); ✓, (6, 10, -1))  
 (até igualAB faça(incA; incC); ✓, (6, 10, 0))  
 (incA; incC; até igualAB faça(incA; incC); ✓, (6, 10, 0))  
 (incC; até igualAB faça(incA; incC); ✓, (7, 10, 0))  
 (até igualAB faça(incA; incC); ✓, (7, 10, 1))  
 (incA; incC; até igualAB faça(incA; incC); ✓, (7, 10, 1))  
 (incC; até igualAB faça(incA; incC); ✓, (8, 10, 1))  
 (até igualAB faça(incA; incC); ✓, (8, 10, 2))  
 (incA; incC; até igualAB faça(incA; incC); ✓, (8, 10, 2))  
 (incC; até igualAB faça(incA; incC); ✓, (9, 10, 2))  
 (até igualAB faça(incA; incC); ✓, (9, 10, 3))  
 (incA; incC; até igualAB faça(incA; incC); ✓, (9, 10, 3))  
 (incC; até igualAB faça(incA; incC); ✓, (10, 10, 3))  
 (até igualAB faça(incA; incC); ✓, (10, 10, 4))  
 (✓, (10, 10, 4))