**Funções**

A implementação de funções é muito semelhante à dos procedimentos. A única diferença é que o retorno da função deverá estar no topo da pilha. Uma maneira conveniente de conseguir o efeito descrito, sem modificar a MEPA, é antes da chamada da função (CHPR) reservamos espaço na pilha para retorno com AMEM 1

A := F(E1, E2, ..., EN)

Exemplo:

AMEM 1 // retorno

CRVL E1

CRVL E2

.

.

.

CRVL EN

CHPR L#NrRotulo#

#EndereçoRertorno#

ARMZ A

Se a função tem N parâmetros, então o i-ésimo parâmetro na memória **M** e terá deslocamento – (N+3-i).

O retorno da função é o parâmetro **zero**, pois o deslocamento deve ser negativo. Note que, o valor retornado pela função ficará automaticamente no topo da pilha, devendo ser armazenado em uma variável ou deslocado.

**Questão**: Como poderia ser implementado o comando **Retorne** na função?

Algoritmo **Exemplo4**

variaveis

x inteiro;

funcao fibo(inteiro a): inteiro

inicio

se (a=0 **e** a=1) entao

fibo := a;

senao

fibo := fibo(a-1) + fibo(a-2);

fim se

fim

inicio

leia (x);

x := fibo (x);

imprima (x)

fim

Encontro os endereços para os parâmetros e o retorno

a = -(1+3-1) = -3

Fibo = -(1+3-0) = -4

**Traduzir para a MEPA o exemplo4 com funções**

**Exercícios1:** Faça a tradução do algoritmo abaixo para MEPA

Algoritmo **ExFuncoes1**

variaveis

x inteiro;

funcao result(inteiro n): inteiro

inicio

se (n=1) entao

result:= 2;

senao

result := 2 \* result(n-1);

fim se

fim

inicio

leia (x);

x := result(x);

imprima (x)

fim

**Exercícios2:** Considere que você já definiu como funciona o **Retorne** na sua linguagem, faça a tradução do algoritmo abaixo para MEPA

Algoritmo **ExFuncoes2**

variaveis

x inteiro;

funcao recursao(inteiro n): inteiro

inicio

se (n<=10) entao

retorne n\*2;

senao

retorne recursao(recursao(n/3));

fim se

fim

inicio

imprima (recursao(27));

fim

**Considere o algoritmo a seguir:**

algoritmo **exemplo5**

variaveis

inteiro n;

procedimento fat(inteiro f; inteiro ref r)

inicio

se (f=1) entao

r:=f;

senao

fat(f-1,r);

r:=f\*r;

fim se

fim

inicio

fat(2,n);

imprima(n);

fim

Parâmetros passados por referência

CREN m,n (carregar endereço)

s <- s+1

M[s] <- D[m]+n

CRVI m,n (Carrega valor indiretamente)

s <- s+1

M[s] <- M[M[D[m]+n]]

ARMI m,n (armazena indiretamente)

M[M[D[m]+n]] <- M[s]

s <- s-1