

- (1) 1. Qual é a importância da Symbol Table durante o desenho do Registo de Activação?
(escolha a hipótese mais correcta)

- A. A Symbol Table contém informação sobre os tipos das variáveis;
- B. A Symbol Table contém informação sobre os tipos das variáveis e das funções;
- C. A Symbol Table contém o tamanho e offset de cada variável;
- D. A Symbol Table contém o endereço do frame pointer;
- E. A Symbol Table não tem importância no desenho do Registo de Activação.

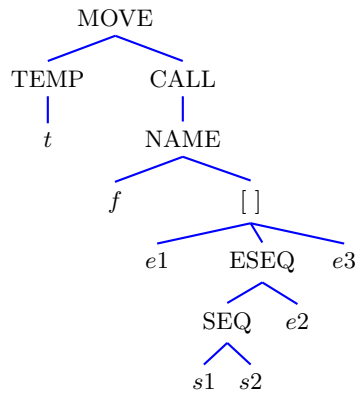
2. Considere o seguinte programa em *Yal!*:

```
1  da(x : int[10], y : int[10]) : void {
2    t : int = 2;
3    i : int = 0;
4
5    while i < 10 do {
6      y[i] = x[i] * t;
7      i = i + 1;
8    };
9  };
10
11 main() : void {
12   i : int = 0;
13   a, b : int[10];
14
15   while i < 10 do {
16     a[i] = i;
17     i = i + 1;
18   };
19
20   da(a, b);
21
22   print(b[3]);
23 };
```

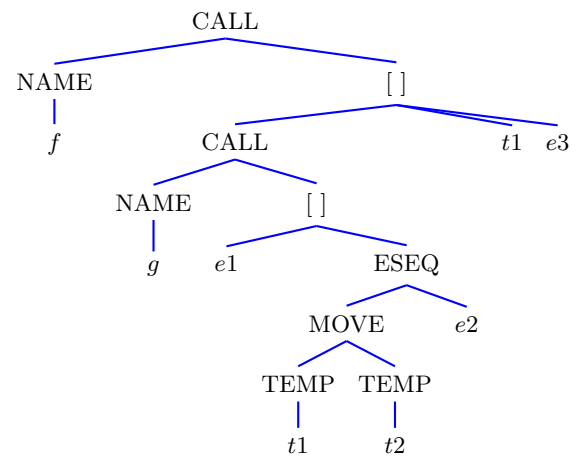
- (1) (a) Resumidamente, o que faz a função **da()**?
- (2,5) (b) Desenhe o registo de activação da função **main()**.
- (2,5) (c) Desenhe o registo de activação da função **da()**.
- (2) (d) Usando o esquema de geração de código para máquina de pilha com instruções MIPS estudado nas aulas, proponha um padrão para geração de código para ciclos **while**.
(pretende-se a definição para a função **codegen()** de um nó **while** da APT)
- (2,5) (e) Como seria o código gerado para a linha 20 (**da(a, b)**)?
- (1,5) (f) Quantos temporários são necessários para o ciclo **while** da função **da()**? (apresente os cálculos que efectuar)

- (3) 3. Utilizando as regras de reescrita para árvores canónicas, proponha formas optimizadas para as seguintes árvores de Representação Intermédia:

(a)



(b)



- (2) 4. Para o programa do exercício (2), proponha uma Representação Intermédia para a linha 6.
(y[i] = x[i] * t)

5. Considere, numa determinada representação intermédia, o seguinte excerto de código:

```

1 L1:
2   MOVE t1 a
3   MOVE a t1
4   MOVE t1 CONST(1)
5   JUMPIFZERO t1 L1
6 L2:
7   GOTO L3
8   MOVE t2 MEM(a)
9 L3:
10  CALL L4 t1 t2
11  ADD t3 t3 CONST(0)
12  SUB t1 t3 t2
13  JUMP L2
  
```

- (2) (a) Proponha optimizações para o código apresentado.