

Considere o seguinte programa em *Ya!*:

```
1  r : int = 1;
2  n : float = 2.5;
3
4  f (n: int) : int {
5      a : int = r * n;
6      r = a - 1;
7
8      return a;
9  };
10
11 main () : void {
12     a,b : int = 0;
13     c[10] : int;
14
15     while a < 10 do {
16         print("Introduza um inteiro:");
17         input(b);
18
19         c[a] = f(b);
20         a = a + 1;
21     };
22
23     print(c);
24 };
```

- (2) 1. Usando a representação abstracta sugerida nas aulas, desenhe a APT resultante da análise do programa.
- (2) 2. Explique, de forma sucinta, como se processa a análise semântica da função `main()`.
- (2) 3. Mostre uma representação do conteúdo da Symbol Table, quando a análise semântica chega ao nó da APT correspondente à linha 17 (exclusive).
4. Proponha um desenho (“esqueleto”) para o Registo de Activação:
 - (1,5) (a) da função `f()`;
 - (2) (b) da função `main()`;
- (2) 5. Considerando que as variáveis `r` e `n` são globais e, por isso, não pertencem a nenhum registo de activação específico, proponha uma forma para permitir o acesso a esse tipo de variáveis.
6. Considere agora o mesmo programa, mas durante a sua execução. Suponha que estamos a meio do ciclo `while` da função `main()` e a variável `a` tem o valor 2.
 - (1,5) (a) Mostre o conteúdo do registo de activação nesse momento da execução do programa.
 - (1,5) (b) A stack irá crescer, diminuir ou manter-se, considerando que o ciclo irá continuar a executar? (Justifique a sua resposta.)

7. Considere a árvore de Representação Intermédia definida pela seguinte expressão:

```
ESEQ( MOVE( MEM( BINOP(+, TEMP(fp), CONST(4) ) ), CALL( NAME(f), [ TEMP(x), ESEQ(
MOVE( TEMP(x), CALL(NAME(g), [TEMP(x)])), TEMP(x) ), BINOP(+, TEMP(x), CONST(1) ))]
), MEM( BINOP( +, TEMP(fp), CONST(4) ) ) ) )
```

- (1,5) (a) Apresente uma instrução em *Ya!* que possa ter resultado nesta representação intermédia.
- (2) (b) Usando as regras de reescrita para árvores canónicas, mostre uma forma optimizada da árvore apresentada.

8. Suponha que vai implementar uma extensão à linguagem *Ya!*: **classes**. Considere o seguinte exemplo:

```
1 . . .
2 pessoa : class(nome: string, idade : int);
3
4 pessoa.nome = ``Ana``;
5 pessoa.idade = 17;
6 . . .
```

- (2) (a) Quais são as implicações desta extensão, relativamente à análise semântica? Dê um exemplo de como se processaria a análise semântica do excerto de programa apresentado.