

1. Considere o seguinte programa em uma linguagem semelhante ao C:

```
main ( )
{
    int a, b ;
    a = 10;
    b = 2;
    soma ( a, b ) ;
}
soma (int x, int y)
{
    int z;
    z = x + y ;
    printf("%d \n", z) ;
}
```

Determine o momento em que ocorrem as alocações e liberações de memória, as amarrações de tipo e de valor das variáveis a, b e z, de acordo com as fases abaixo.

- Fase 1 - Compilação
- Fase 2 - Do início da execução do programa (execução de main, incluindo elaboração) até a chamada do procedimento soma;
- Fase 3 – Durante a elaboração do procedimento soma;
- Fase 4 - Durante a execução do procedimento soma;
- Fase 5 – No retorno do procedimento soma;
- Fase 6 - Do retorno do procedimento soma até o final da execução do programa.

Amarrações de valor

a – fase 2

b – fase 2

z – fase 4

Amarrações de tipo

a – fase 1

b – fase 1

z – fase 1

Alocação e liberação de memória

a – alocação fase 2, liberação fase 6

b – alocação fase 2, liberação fase 6

z – alocação fase 3, liberação fase 5.

2. Considere o seguinte programa, escrito em uma linguagem tipo PASCAL:

```
program main;
var x, y, z : integer;
procedure sub1;
    var a, y, z : integer;
    begin {sub1}
        .....
    end; {sub1}
procedure sub2;
    var a, b, z : integer;
    begin {sub2}
        .....
    end; {sub2}
procedure sub3;
    var a, x, w : integer;
    begin {sub3}
        .....
    end {sub3}
begin { main }
.....
end. { main }
```

Para a sequência de chamadas: main chama sub1; sub1 chama sub2; sub2 chama sub3, diga quais variáveis (locais e não locais) são visíveis durante a execução de sub3, considerando:

- a) amarração de escopo dinâmica

main: ~~x~~, ~~y~~, ~~z~~ → sub1: ~~a~~, y, z → sub2: ~~a~~, b, z → sub3: a, x, w
a, x, w de sub3, b, z de sub2, y de sub1.

- b) amarração de escopo estática

a, x, w de sub3, y, z de main

3. Considere o seguinte programa escrito em uma linguagem semelhante ao Pascal. Supondo regras de escopo estático, qual valor de x é impresso no procedimento sub1? E para regras de escopo dinâmico?

```
program main;
var x : integer;
procedure sub1;
begin { sub1 }
  writeln ( 'x = ', x)
end; { sub1 }
procedure sub2;
var x : integer;
begin { sub2 }
  x := 10;
  sub1
end; { sub2 }
begin { main }
  x := 5;
  sub2
end. { main }
```

Escopo estático: x = 5

Escopo dinâmico: x = 10

4. Considere o seguinte programa C esquemático:

```
void fun1 (void);
void fun2 (void);
void fun3 (void);
void main ( ) {
    int a, b, c;
    ....
}
void fun1 (void) {
    int b, c, d;
    ...
}
void fun2 (void) {
    int c, d, e;
    ...
}
void fun3 (void) {
    int d, e, f;
    ...
}
```

Dadas as seguintes seqüências de chamadas e supondo que seja usado o escopo dinâmico, quais variáveis são visíveis durante a execução da última função chamada? Diga, para cada variável visível na última função, o nome da função em que ela foi declarada.

- main chama fun1; fun1 chama fun2; fun2 chama fun3.

main: a, ~~b~~, ~~c~~ → fun1: b, ~~c~~, ~~d~~ → fun2: c, ~~d~~, ~~e~~ → fun3: d, e, f

Variáveis visíveis: d, e, f de fun3, c de fun2, b de fun1, a de main

- main chama fun2; fun2 chama fun3; fun3 chama fun1.
 main: a, b, c → fun2: c, d, e → fun3: d, e, f → fun1: b, c, d
 Variáveis visíveis: b, c, d de fun1, e, f de fun3, a de main

5. Para o programa dado a seguir, em uma linguagem tipo PASCAL, diga quais variáveis (de quais procedimentos) estão sendo usadas nos comandos numerados, para os seguintes casos:

- variável D no ponto 1, variável B nos pontos 2 e 3 considerando escopo estático;
- variável D no ponto 1, variável B nos pontos 2 e 3 considerando escopo dinâmico.

Note que a linguagem usada permite aninhamento de subprogramas.

```

program MAIN;
  var A, B, C : integer;
  procedure SUB1 (X : integer) ;
    var A, D : integer;
    procedure SUB4;
      begin {SUB4 }
        .....
        A := D / 2; -----→ 1
        .....
      end; { SUB4 }
    begin { SUB1}
      .....
      D := X + 1;
      .....
      SUB4;
      .....
      B := A + X; -----→ 2
      .....
    end; { SUB1}
  procedure SUB2;
    var B, E : integer;
    procedure SUB3;
      var C, E : integer;
      begin { SUB3 }
        B := 0;
        SUB1( 5);
        .....
        E := B + A;
      end ; {SUB3}
    begin { SUB2 }
      .....
      SUB3;
      .....
      A := B + 1; -----→ 3
      ....
    end; { SUB2 }
  begin { MAIN }
    A := 100;
    .....
    SUB2;
    ....
  end; { MAIN }

```

Escopo estático:

Ponto 1 – D de SUB1

Ponto 2 – B de MAIN

Ponto 3 – B de SUB2

Escopo dinâmico:

Ponto 1 – D de SUB1

Ponto 2 – B de SUB2

Ponto 3 – B de SUB2

6. Considerando o mesmo programa do exercício anterior, determine o momento (fases, semelhantes as do exercício 1) em que ocorrem as alocações e liberações de memória, as amarrações de tipo e de valor de todas as variáveis que aparecem no programa, inclusive parâmetros.
7. Considere o programa dado a seguir, escrito em uma linguagem tipo Pascal (que permite subprogramas aninhados). Determine quais variáveis estão com memória alocada a elas e quais são visíveis no ponto 1 indicado no código, considerando amarração de escopo estático.

```

program MAIN_1;
  var P : real;
  procedure A(X : integer);
    var Y : boolean;
    procedure C(Q : boolean);
      begin { C }
      ..... -----> 1
    end; { C }
  begin { A }
  .....
  C (Y);
  .....
  end; { A }
  procedure B (R : real) ;
    var S, T : integer;
    begin { B }
    .....
    A (S);
    .....
    end; { B }
  begin { MAIN_1 }
  .....
  B (P);
  .....
end. { MAIN_1 }

```

Variáveis com memória alocada no ponto 1:

Variável P de MAIN_1

Parâmetro R de B

Variáveis S e T de B

Parâmetro X de A

Variável Y de A

Parâmetro Q de C

Variáveis visíveis no ponto 1:

Parâmetro Q de C

Variável Y de A

Parâmetro X de A

Variável P de MAIN_1