

BÀI TẬP**ĐỊNH DANH, PHẠM VI VÀ BINDING**

Bài 1. Cho mã giả của một chương trình như sau:

```

procedure main
  a: integer := 1
  b: integer := 2
  procedure middle
    b: integer := a
    procedure inner
      print a, b
      a: integer := 3
      inner()
      print a, b
    middle()
  print a, b

```

- Giả sử đoạn code nói trên tuân theo qui luật thứ tự khai báo của C, nghĩa là định danh phải được khai báo trước khi sử dụng và phạm vi của nó tính từ lúc khai báo đến hết khối lệnh (block). Cho biết kết quả được in ra màn hình
- Cùng câu hỏi trên, nhưng tuân theo qui luật thứ tự khai báo của C#, nghĩa là định danh phải được khai báo trước khi sử dụng và phạm vi của nó là toàn bộ khối lệnh mà khai báo đó xuất hiện.

Bài 2. Giả sử chương trình Javascript sau được thông dịch và sử dụng qui tắc phạm vi tĩnh. Hãy cho biết giá trị của x trong hàm sub1? Nếu là phạm vi động thì giá trị của x trong hàm sub1 là bao nhiêu?

```

var x;
function sub1( ) {
  document.write ("x = " + x + "<br />");
}
function sub2( ) {
  var x;
  x = 10;
  sub1();
}
x = 5;
sub2( );

```

Bài 3. Cho chương trình C như sau:

```
void fun1( void ) /* prototype */
void fun2( void ) /* prototype */
void fun3 (void ) /* prototype*/
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;
}
```

Cho các chuỗi gọi hàm như bên dưới, giả sử rằng việc qui tắc của phạm vi động được sử dụng. Hãy cho biết các biến nào là có thể tham chiếu được trong suốt thời gian hàm cuối cùng thực thi

- main gọi fun1; fun1 gọi fun2; fun2 gọi fun3
- main gọi fun1; fun1 gọi fun3
- main gọi fun2; fun2 gọi fun3; fun3 gọi fun1
- main gọi fun3; fun3 gọi fun2; fun2 gọi fun1

Bài 4. Cho một đoạn chương trình viết bằng ngôn ngữ C như sau:

```
{
    int a, b, c;
    ...
    {
        int d, e;
        ...
        {
            int f;
            ...
        }
        ...
    }
}
```

```

...
{
    int g, h, i;
    ...
}
...
}

```

Giả sử mỗi biến kiểu int chiếm 4 bytes. Cho biết tổng không gian bộ nhớ (theo byte) cần cho các biến trong đoạn code trên.

Bài 5. Cho đoạn mã giả như sau:

```

procedure P ( A, B : real )
    X : real
    procedure Q (B, C : real)
        Y : real
        ...
    procedure R (A, C : real)
        Z: real
        ... (*)
    ...

```

Giả sử sử dụng phạm vi tĩnh, cho biết môi trường tham chiếu ở vị trí đánh dấu (*)

Bài 6. Cho mã giả của một chương trình như sau, giả sử cho phép thủ tục lồng nhau và sử dụng phạm vi tĩnh

```

procedure main
    g: integer
    procedure B( a : integer)
        x: integer
        procedure A( n: integer)
            g := n
        procedure R (m : integer)
            write_integer( x )
            x /= 2 // chia số nguyên
            if x > 1
                R (m + 1)
            else
                A (m)
            x := a x a
            R (1)
        B (3)
    write_integer(g)

```

- a. Cho biết chương trình in kết quả gì
- b. Vẽ các frame trên ngăn xếp khi hàm A vừa được gọi.