

Câu 1. Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1
```

Pascal -> static scope

```
    procedure a(x:real) //line 2
```

```
        var z: integer; // line 3
```

```
            procedure b(y:integer) // line 4
```

```
                var z: real; // line 5
```

```
                begin... end //of b
```

```
            begin... ... end // of a
```

```
begin... end
```

Trong các câu sau, để đảm bảo so trùng, viết các tên tăng dần theo thứ tự alphabet và cách nhau bằng dấu phẩy, không có khoảng trắng. Chỉ kể các tên có trong đoạn chương trình trên, nếu có nhiều tên trùng nhau (như x,y,z) phải viết kèm dấu // và số line của khai báo, ví dụ x//1 hoặc y//4.

Môi trường tham khảo **cục bộ** của thủ tục b gồm có các tên y//4, z//5

Môi trường tham khảo không cục bộ của thủ tục b gồm có các tên a, b, x//2

Môi trường tham khảo toàn cục của thủ tục b gồm có các tên a

Đây là các tên được khai báo ở phạm vi ngoài cùng nhất (global scope) mà b có thể truy cập.

Giải

- Môi trường tham khảo cục bộ của thủ tục b gồm có các tên y//4, z//5
- Môi trường tham khảo không cục bộ của thủ tục b gồm có các tên a, b, x//2,
- Môi trường tham khảo toàn cục của thủ tục b gồm có các tên: a.

Câu 2. Cho đoạn mã C++ sau, hãy cho biết các đối tượng z và new int được cấp phát trong các vùng nhớ nào?

```
int * m;
```

```
int * foo(int x) {
```

```
    static int y;
```

```
    int * z = new int;
```

```
    switch x {
```

```
        case 1: return &y;
```

```
        case 2: return &x;
```

```
        case 3: return z;
```

```
        case 4: return m;
```

```
    }
```

```
}
```

Giải:

x được cấp phát trong vùng nhớ **stack** vì là biến cục bộ trong hàm foo ; new int được cấp phát trong vùng nhớ **heap** do được cấp phát động.

Câu 3. Cho đoạn mã C++ sau

```
int * m;  
  
int * foo(int x) {  
    static int y;  
    int * z = new int;  
    switch x {  
        case 1: return &y;  
        case 2: return &x;  
        case 3: return z;  
        case 4: return m;  
    }  
}
```

Hãy điền tên của biến thích hợp vào các chỗ trống sau:

Chú ý:

- Chỉ điền tên 1 biến vào ô trống ngay cả khi có nhiều tên biến thích hợp để điền
- Nếu cần điền đối tượng được tạo ra bởi new int thì ghi là new int (7 ký tự bao gồm 1 khoảng trắng)

Đối tượng____(1)____ có thời gian sống bằng với biến z.

Đối tượng____(2)____ có thời gian sống bằng thời gian thực thi của hàm foo.

Đối tượng____(3)____ có thời gian sống bằng với biến m.

Đối tượng____(4)____ có thời gian sống dài hơn thời gian thực thi của hàm foo nhưng không bằng thời gian thực thi của chương trình chính.

Giải:

- (1) là x ; (2) là x hoặc z ; (3) là y ; (4) là new int.

Thứ tự thực thi của hàm foo()

Câu 4. Cho biết trong đoạn mã C++ sau, đối tượng nào có nhiều tên và những tên này là gì?

```
int *p = new int;
```

```
int *q=p;
```

A. Đối tượng có nhiều tên là đối tượng được tạo ra bởi lệnh "new int" và các tên của đối tượng này là p và q

B. Đối tượng có nhiều tên là p và các tên của nó là p và q

C. Đối tượng có nhiều tên là đối tượng được tạo ra bởi lệnh "new int" và các tên của đối tượng này là *p và *q

D. Không có câu nào đúng

Giải: Chọn C.

Đối tượng có nhiều tên là new int. 2 tên của nó là *p và *q.

Câu 5. Cho phát biểu C++ sau:

`x = x + y * 10;`

Trong biểu thức bên vế phải, phép nhân * được tính toán trước phép +. Ràng buộc này xảy ra vào thời gian nào?

A. Hiện thực ngôn ngữ (Language implementation time) B. Thời gian thực thi (Runtime)

C. Thời gian dịch (Compiling time)

D. Định nghĩa ngôn ngữ (Language design time)

Giải.

Chọn D. Thứ tự ưu tiên toán tử được quy định trong giai đoạn định nghĩa ngôn ngữ.

Câu 6. Cho phát biểu Python sau:

`x = x * y** 3;`

Trong biểu thức bên vế phải, phép nhân được tính toán sau phép ** ; ràng buộc này xảy ra vào thời gian nào?

Chọn một:

A. Thời gian dịch (Compiling time)

B. Hiện thực ngôn ngữ (Language implementation time)

C. Định nghĩa ngôn ngữ (Language design time)

D. Thời gian thực thi (Runtime)

Giải: Chọn C . Thứ tự ưu tiên toán tử được quy định trong giai đoạn định nghĩa ngôn ngữ.

Hiện thực ngôn ngữ: Assignment 1, 2, 3 ,4.

Thời điểm dịch:

Câu 7. Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal

```
var x,y:integer; // line 1 (x, y toàn cục)

procedure foo(x:real) // line 2 (foo được
định nghĩa ở phạm vi toàn cục)
  var z: integer; // line 3 (z cục bộ của
foo)
begin ... end // of foo

procedure foo1(y:integer) // line 4 (foo1
được định nghĩa ở phạm vi toàn cục)
  var z: real; // line 5 (z cục bộ của foo
1)
begin ... end // of foo1

begin (* main program *)
...
end.
```

```
var x,y:integer; // line 1

procedure foo(x:real) //line 2

var z: integer; // line 3

begin .... end

procedure foo1(y:integer) // line 4

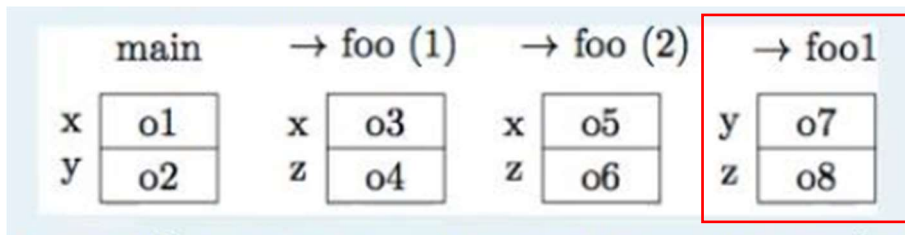
var z: real; // line 5

begin ... end

begin ... end
```

static scope

Giả sử chương trình chạy theo trình tự: main gọi foo, foo gọi đệ qui foo, foo gọi foo1 với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



foo, foo1 và main đều nằm cùng cấp với nhau, x và y được khai báo global. → foo1 thấy được y, z của chính nó và x global.

Hãy cho biết môi trường tham khảo động của foo1 có chứa các đối tượng nào

A. o1 B. o2 C. o3 D. o4 E. o5 F. o6 G. o7 H. o8

Giải: Chọn A, G, H. Ngôn ngữ này là ngôn ngữ có tầm vực tĩnh

x → o1

y → o7

z → o8

Câu 8. Cho đoạn chương trình sau viết trên một ngôn ngữ tầm vực động.

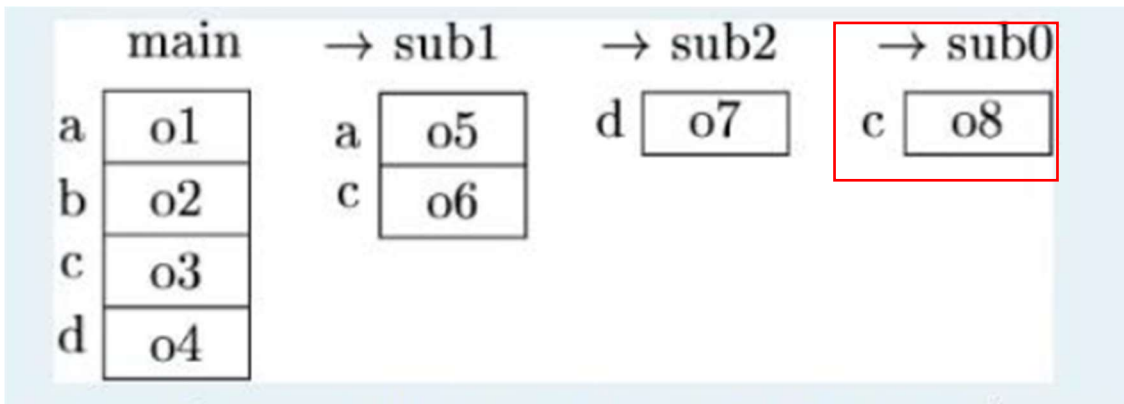
```
program main;
var a,b,c,d: integer; // line 1
begin ... end

procedure sub0
var c: integer; // line 2
begin .. end;

procedure sub1(a:real) //line 3
var c:real; //line 4
begin ... end

procedure sub2(d:char) //line 5
begin ... end
```

Giả sử chương trình chạy theo trình tự: main gọi sub1, sub1 gọi sub2, sub2 gọi sub với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



Hãy cho biết môi trường tham khảo động của sub0 có chứa các đối tượng nào.

A. o1 B. o2 C. o3 D. o4 E. o5 F. o6 G. o7 H. o8

Giải. Chọn B, E, G, H.

c->o8

d->o7

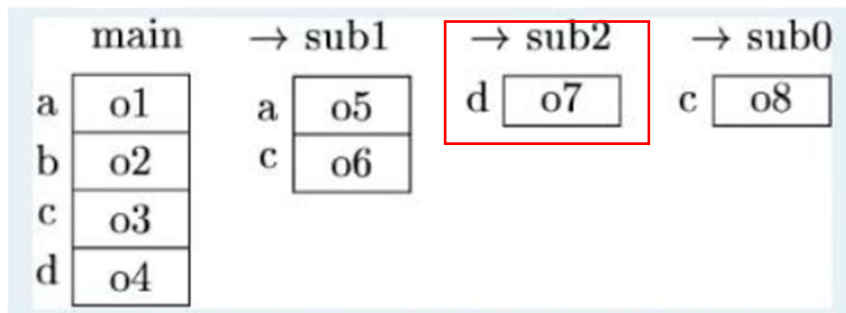
a->o5

b->o2

Câu 9. Cho đoạn chương trình sau viết trên một ngôn ngữ tầm vực động.

```
program main;  
var a,b,c,d: integer; // line 1  
begin ... end  
  
procedure sub0  
var c: integer; // line 2  
begin .. end;  
  
procedure sub1(a:real) //line 3  
var c:real; //line 4  
begin ... end  
  
procedure sub2(d:char) //line 5  
begin ... end
```

Giả sử chương trình chạy theo trình tự: main gọi sub1, sub1 gọi sub2, sub2 gọi sub với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



Hãy cho biết môi trường tham khảo động của sub2 có chứa các đối tượng nào.

A. o1 **B. o2** C.o3 D.o4 **E.o5** **F.o6** **G.o7** H.o8

Giải: Môi trường tham khảo động của sub2 gồm o7, o5, o6, o2

d->o7 ; a->o5, c->o6, b->o2.

Câu 10. Cho một đoạn chương trình sau viết trên một ngôn ngữ tựa khối cấu trúc như Pascal

```

var x,y:integer; // line 1

procedure foo(x:real) //line 2

var z: integer; // line 3

begin .... end

procedure foo1(y:integer) // line 4

var z: real; // line 5

begin ... end

begin ... end

```

static scope

x, y

```

foo(){
  x, z
}

```

```

foo1(){
  y, z
}

```

```

main(){
}

```

Hãy cho biết các khai báo có hiệu lực trong procedure foo

Chọn một hoặc nhiều hơn.

A. Khai báo x ở line 2

B. Khai báo foo ở line 2

C. Khai báo x ở line 1

D. Khai báo y ở line 1

E. Khai báo z ở line 3

F. Khai báo y ở line 4

G. Khai báo z ở line 5

H. Khai báo fool ở line 4

Giải: Chọn A, B, D, E, H

foo, foo1, x//2, z//3, y//1

Question 4

Not yet answered

Marked out of 1.00

Flag question

Cho đoạn mã C++ sau, hãy cho biết khi gọi hàm foo với thông số là 3 thì sẽ có lỗi gì (trong các lỗi tham chiếu treo, garbage hoặc alias) xảy ra?

```

int * m;

int * foo(int x) {
    static int y;
    int * z = new int;
    switch x {
        case 1: return &y;
        case 2: return &x;
        case 3: return z;
        case 4: return m;
    }
}

```

Select one:

- ☐ a. Gây ra lỗi tạo ra rác
- ☐ b. Không gây ra lỗi gì cả
- ☐ c. Gây ra alias
- ☐ d. Gây ra lỗi tham chiếu treo (dangling reference)

Time left 0:06:37

Giải: Chọn B.

Trong trường hợp case 3: return z;, vùng nhớ mà z trỏ tới vẫn còn hợp lệ sau khi hàm foo kết thúc vì nó được cấp phát trên heap và chưa bị delete

Alias: Tên gọi thứ n cho cùng 1 đối tượng.

Dangling reference: Ô nhớ được cấp phát trong vùng nhớ heap, có địa chỉ nhưng chưa đc hủy cấp phát.

Rác: Tham chiếu đến một ô nhớ không còn tồn tại.

Khi return z, thứ được trả về là địa chỉ đến ô nhớ được cấp phát trong vùng nhớ heap. Ô nhớ này sẽ chỉ bị xoá khi gọi toán tử delete.

Question

7

Not yet
answered

Marked out of
1.00

Flag question

Cho đoạn mã C++ sau:

```
int s = 0;
```

```
int *p = &s;
```

```
int *q = p;
```

```
int r = *p;
```

Nhắc lại tác vụ &s trên C++ trả về địa chỉ của biến s, *p trả về giá trị nguyên của đối tượng mà p trỏ đến.

Hãy viết theo thứ tự alphabet các alias được tạo ra trong đoạn code trên?

Để đảm bảo trùng khớp, các tên phải được viết:

- cách nhau bằng dấu phẩy, không có khoảng trắng,
- theo thứ tự alphabet (ví dụ p,q)
- nếu có dấu *, & thì các dấu này viết trước ký tự (ví dụ *r,p), dấu * viết trước & (ví dụ *s,&r)

Answer:

Giải: Các tên đó là s, *p, *q

Alias cũng có thể là tên của chính nó, phải không nè :v