**# Bài 1**

def mul(root):

if (root.val == 0 or root.letf.val == 0 or root.right.val == 0):

return 0

else: return root.val\*mul(root.left)\*mul(root.right)

**# Bài 2:**

void swap(int x, int y) {

int t = x;

x = y;

y = t;

}

void main() {

int a[] = {2, 1, 0};

i = 0;

swap(i, a[i]); // swap in main

cout << i << a[0] << a[1] << a[2];

}

A. x,y are passed by value

-> Output: 0210

Giải thích: vì x,y được truyền vào là giá trị, tức là bản coppy của gía tri thực tế, caller

và callee là hai biến độc lập nhưng có cùng giá trị, nếu callee thay đổi giá trị thì caller

không đổi giá trị

B.x,y are pass by value-result

-> Output: 2010

Giải thích: pass by value-result bao gồm truyền giá trị khi chương trình con được gọi, tuy

nhiên điểm khác biệt so với truyền bằng giá trị là khi chương trình con kết thúc, nó gán

giá trị hiện tại ngược lại cho tham số chính ban đầu(formal parameter)

C. x,y are pass by reference

-> Output: 2010

Giải thích: gía trị truyền vào là địa chỉ của biến cái mà lưu trữ giá trị của date trên bộ nhớ,

callee đưa trực tiếp địa chỉ cho caller, chính vì thế nó làm thay đổi giá trị trên giá tri gốc

D. x,y are passed by name

-> Output: 2210

Giải thích: khi gọi hàm swap(int x, int y), x được xem như là 1 bí danh mới của i, y được

xem như là một bí danh mới của a[i].

void swap(int x, int y) { biến i có tên mới là x, biến a[i] có tên mới là y

int t = x; // t = i (i = 0 -> t = 0)

x = y; // i = a[i] hay i = a[0] -> i = 2

y = t; // a[i] = t mà i = 2, t = 0 -> a[2] = 0

}

**# Bài 3:**

Trong pascal, sử dụng 'var' là truyền bằng tham khảo nghĩa là truyền bởi con trỏ.

P := @A -> P giữ địa chỉ của mảng A

Q := @B -> Q giữ địa chỉ của mảng B

A. at line //1

-> R^[1]:= R^[1] + 10; //1

Vì SUB2(P,Q) mà P truyền vào là passed by value, p := @A nên sau khi gọi hàm

thì giá trị của A vẫn không đổi, giá trị của R^[1] = R^[1] + 10 = A[1] + 10 = 17;

B. at line //2

S^[1] := S^[1] + 10; // 2

-> Vì S được truyền vào là passed by reference mà S đang giữ địa chỉ của B, nên

sau khi gọi thủ tục SUB2 giá trị của B sẽ đươc cập nhật thành 17 (S^[1] = S^[1] + 10 = B[1] + 10 = 17;)

C. at line //3

if ... then R := S ; // 3

-> Giá trị của R sẽ thay đổi thành 17 tuy nhiên A[1] vẫn không đổi

D. at line //4

else S := R; // 4

-> Giá trị của B[1] sẽ bị thay đổi thành 17 vì S đươc truyền passed by reference

**# Bài 4:**

n := 5 ;

for i := 1 to n do begin

n := n + 1 ;

i := i − 1 ;

end

A. expr2 is protected

-> For loop thưc thi vô hạn, vì expr2 là protected nên giá trị của n sẽ không thể nào thay đổi

và luôn giữ giá trị là 5, i sau mỗi loop sẽ giảm đi 1 lần, nên i sẽ không thể nào chạm được n

B. index is protected

-> For loop sẽ lặp vô hạn lần vì giá tri của i sẽ không đổi, trong khi đó n sau mỗi lần

lặp sẽ tăng lên 1, nên n và i không thể chạm nhau

C. no protect

-> For loop sẽ vô hạn lần vì cả hai giá tri sẽ thay đổi và càng tách xa nhau

D. both index and expr2 are protected

-> For loop sẽ lặp 4 lần(hoặc 5 tuỳ theo ngôn ngữ) vì sau mỗi vòng for n và i không đổi giá trị (index và expr2 là protected)