

Question 1.

def mul(root):

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

return 0

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

Question 2.

a) x,y are passed by value

Output: 0210

Giải thích: Vì ta truyền theo kiểu value thì khi truyền vào giá trị bao nhiêu thì các parameter tương ứng thứ tự nào sẽ nhận giá trị theo thứ tự đó.

b) x,y are passed by value-result

Output: 2010

Giải thích: Vì khi ta truyền theo kiểu value –result thì parameter được truyền nếu thay đổi giá trị thì nó sẽ gán giá trị hiện tại ngược lại cho tham số chính ban đầu

c) x,y are passed by reference

Output: 2010

Giải thích: Vì khi ta truyền theo kiểu Reference thì ta sẽ truyền địa chỉ của biến vào parameter của hàm gọi nên khi thay đổi parameter cũng đồng nghĩa thay đổi giá trị trên biến tại địa chỉ mà parameter đang nhận.

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  
}
```

Question 3.

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

$P := @A \rightarrow$ P giữ địa chỉ của mảng A

$Q := @B \rightarrow$ Q giữ địa chỉ của mảng B

a) at line //1

$R^{\wedge}[1] := R^{\wedge}[1] + 10; //1$

P được truyền vào là pass by value, $P := @A$ nên sau khi gọi hàm thì giá trị của A vẫn không đổi.

Chỉ có giá trị $R[1]$ thay đổi là $R^{\wedge}[1] = R^{\wedge}[1] + 10 = A[1] + 10 = 17;$

b) at line //2

$S^{\wedge}[1] := S^{\wedge}[1] + 10; //2$

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 được cập nhật thành $S^{\wedge}[1] = S^{\wedge}[1] + 10 = B[1] + 10 = 17;$

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 được truyền passed by reference

Question 4.

a) expr2 is protected

For loop thực 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 trị 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 trị sẽ thay đổi và càng tách xa nhau

d) both index and expr2 are protected

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

