Khong Thi Bich Ngoc – 1727029 – Tutorial

**1.**

int mul (Node root ) {

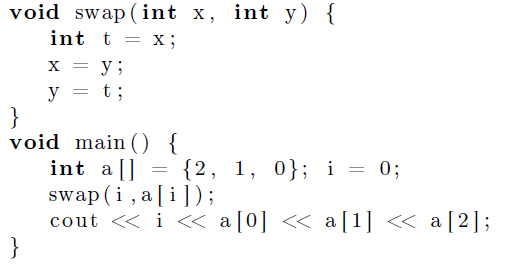
if ( root == null ) return 1 ;

if (root.val == 0 || mul(root.left) == 0 || mul(root.right) == 0) return 0;

return root.val \* mul(root.left) \* mul(root.right);

}

**2.**



**A. x,y are passed by value**

After calling swap(i,a[i]) and returning from it, values of i and a[i] do not change

Print: 0210

**B. x,y are passed by value-result**

When calling swap(i,a[i]):

i = 0, a[i] is a[0] = 2

Beginning: x receives value 0, y receives value 2

t = x -> t = 0

x = y -> x = 2

y = t -> y = 0

Returning from it, i receives value 2 and a[0] receives value 0

Print: 2010

**C. x,y are passed by reference**

When calling swap(i,a[i]):

i = 0, a[i] is a[0] = 2

Beginning: x is new name for i, y is new name for a[0]

t = x -> t = 0

x = y -> x = 2 -> i = 2

y = t -> y = 0 -> a[0] = 0

Returning from it, i keeps the new value 2, a[0] keeps the new value 0

Print: 2010

**D. x,y are passed by name**

Pass by name: similar to pass by ref, but lazy evaluation, only evaluate parameters when they are actually used

When calling swap(i,a[i]):

i = 0, a[i] is a[0] = 2

Beginning: x is new name for i, y is new name for a[0]

t = x -> t = 0

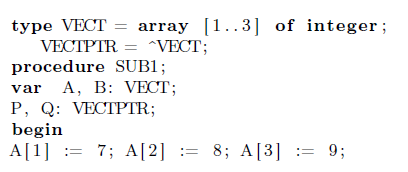
x = y -> x = 2 -> i = 2

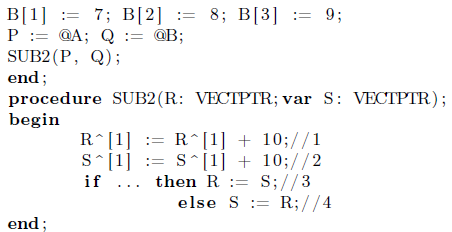
y = t -> y = 0 -> a[i] = a[2] = 0

Returning from it, i keeps the new value 2, a[2] keeps the new value 0

Print: 2210

3.





Which is **object** changed by the following assignment and what is the value assigned?

**A. at line //1**

In SUB1: Pointer P contains address of A, pointer Q contains address of Q.

In SUB2:

Formal parameters R, S are declared as pointers, and explicitly dereferenced when they are used

R is passed value of P -> reference to A

S is passed value of Q -> reference to B

R^ -> is dereferenced to obtain the value it refers to

* Object changed is A[1]

A[1] = A[1] + 10 -> A[1] = 17

**B. at line //2**

With above explanation

S^ -> is dereferenced to obtain the value it refers to

* Object changed is B[1]

B[1] = B[1] + 10 -> B[1] = 17

**C. at line //3**

Object changed is A

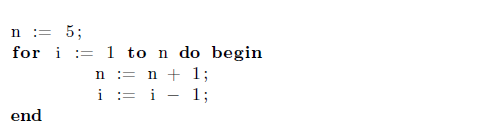
A = B [17,8,9]

**D. at line //4**

Object changed is B

B = A [17,8,9]

**5.**

****

How many iterations does the for loop perform when?

A. expr2 is protected

indefinite

B. index is protected

indefinite

C. no protect

5

D. both index and expr2 are protected

indefinite