

Nguyễn Ngô Thanh Trúc - 1910650

Nguyễn Minh Phú - 1914659

Lê Thanh An - 1912523

Question 1:

int mul (Node root){

if (root == null) return 1;

else if (root.val == 0) return 0;

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

}

Question 2:

0210

Do pass by value nên giá trị thông số thực được sao chép cho thông số hàm, giá trị thông số thực không bị thay đổi nên i và a[i] vẫn giữ giá trị như cũ.

1310

Do pass by value-result nên sau khi kết thúc hàm, giá trị thông số hàm sao chép vào giá trị thực nên làm thay đổi i và a[i]

1310

Do pass by reference nên khi các thông số hàm thay đổi cũng làm thông số thực thay đổi, thông số truyền vào tham khảo đến biến i và a[0]

1310

Do pass by name nên khi các thông số hàm thay đổi cũng làm thông số thực thay đổi, thông số truyền vào là i và a[0]

=======================================================================

Võ Nguyễn Thiện Nhân 1919409

Lê Tiến Dũng - 1811775

Nguyễn Văn Hoàn - 1711376

Question 1:

int mul\_support(Node\* p) {

        try {

            return mul(this->root);

        }

        catch (const char\* mess) {

            return 0;

        }

    }

    int mul(Node \*p)

    {

        if (p == nullptr) return 1;

        else if (p->value == 0) throw 0;

        else

        {

            return p->value \* mul(p->pLeft) \* mul(p->pRight);

        }

    }

Question 2

x, y are passed by value

Kết quả:  0 2 1 0

Giải thích: vì x, y được truyền tham trị nên khi thay đổi giá trị biến x và y trong thân hàm thì nó sẽ không ảnh hưởng đến biến i và a[i]

x, y are passed by value-result

Kết quả: 1 3 1 0

Giải thích: vì x và y được truyền theo kiểu value-result nên khi thay đổi giá trị biến x và y trong thân hàm thì kết thúc hàm sẽ trả ngược lại kết quả cho biến i và a[i]

x, y are passed by reference

Kết quả: 1 3 1 0

Giải thích: vì x, y được truyền tham khảo, nghĩa là địa chỉ biến i được gán cho biến x, địa chỉ của a[i] được gán cho biến y, nên khi thay đổi giá trị biến x và y trong thân hàm thì nó cũng thay đổi giá trị của biến i và a[i]

x, y are passed by name

Kết quả: 1 2 2 0

Giải thích: vì x, y được truyền theo kiểu pass by name, tức là truyền biểu thức vào thông số hình thức nên x++ nghĩa là i++ → i = 1 và y++ nghĩa là a[i]++, lúc này giá trị của i là 1 ⇔ a[1]++ là 2

=======================================================================

**========================================================================**

Nguyễn Duy Kiên - 1812704

Nguyễn Hữu Anh Đại - 2020020

Lê Hồng Phong - 1813518

Q1.

def mul(Node root):

If root == None:

return 1

return 0 if root.val == 0 else root.val \* mul(root.left) \* mul(root.right)

Q2.

x, y truyền bằng giá trị

Giá trị của i và a[i] là 0 và a[0] = 2 được copy vào tham số hình thức x và y

. Sau khi kết thúc hàm increase x = 1, y= 3, do truyền bằng giá trị nên không trả về thông số thực.

Nên i = 0, a = 2.

Màn hình: 0210

x, y truyền theo value result

Giá trị của i và a[i] là 0 và a[0] = 2 được copy vào tham số hình thức x và y

. Sau khi kết thúc hàm increase x = 1, y= 3, do truyền bằng value result nên trả về kết quả cho thông số thực.

Nên x-> i = 1

y-> a[i] = a[1] = 3

Màn hình: 1230

x, y được truyền theo tham chiếu

Địa chỉ của i và a[i] được chuyền vào x và y

Sau khi kết thúc hàm increase x = 1, y = 3, do x và y giữ địa chỉ của i và a[i] nên khi x, y thay đổi sẽ thay đổi i và a[i] theo.

Nên x -> i = 1

y-> a[0] = 3 , do y lưu địa chỉ của biến a[0] chứ không phải là a[i].

Màn hình: 1310

x, y được truyền theo tên

Nên khi thực hiện x++, x được suy diễn ra biến i nên giá trị của i tăng 1 thành 1

Khi thực hiện y++, y được suy diễn từ a[i]. Với i lúc này = 1 suy ra y là a[1] được tăng thêm 1 bằng 2

Vậy khi in ra màn hình được kết quả: i = 1, a[0] = 2, a[1] = 2, a[2] = 0

Màn hình: 1220

========================================================================

Nguyễn Hồng Thiện - 1915290

Trần Tiến Lộc - 1914038

Huỳnh Thành Đạt - 1910110

Question 1:

int mul(Node root) {

try {

return mul\_exception(root);

} catch(int x) {

return 0;

}

}

int mul\_exception(Node root) {

if (root == null) return 1;

else if (root.val == 0) throw 0;

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

}

 => Nếu có node có giá trị là 0, hàm mul\_exception throw giá trị 0. Hàm mul bắt số nguyên 0 (để phân biệt với các exception khác có thể xảy ra), trả về giá trị 0.

Question 2:

Truyền bằng giá trị:

x,y truyền giá trị của i, a[i] là 0, 2

x++; y++; → x = 1, y = 3

Giá trị của i và a[i] không đổi vì truyền bằng giá trị

Kết quả: 0210

Truyền bằng giá trị - kết quả:

x, y được truyền giá trị của i, a[i] là 0, 2

x++; y++; → x = 1; y = 3;

Giá trị x, y được trả về cho i và a[i] sau khi kết thúc chương trình con

Kết quả: 1310

Truyền bằng tham chiếu:

Địa chỉ của i và a[i] được truyền cho x, y → x, y giống với i, a[i]

x++, y++ → x = i = 1; y = a[0] = 3

Kết quả: 1310

Truyền bằng tên

x, y được truyền bằng tên i, a[i] nghĩa là x = i = 0, y = a[i]

x++; y++; → x = i = 1; y = a[1] = 2;

Kết quả: 1220

========================================================================

Các thành viên

Vũ Minh Dương: 1810885

Nguyễn Thọ Nam - 1911650

Lê Ngọc Minh Nhân - 1910402

Câu 1:

int mul (Node root) {

if (root == null) return 1;

try {

if (root.val == 0 ) throw “Is 0”;

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

} catch (...) {

return 0;

}

}

Câu 2:

x, y truyền theo giá trị

Kết quả được in ra là: 0210

Nếu x, y được truyền theo giá trị, có nghĩa x, y chỉ là bản sao của giá trị thực của các tham số được truyền vào, nên nếu có sửa đổi giá trị tham số, thì cũng thay đổi giá trị thực của tham số đó.

x,y truyền theo giá trị - trả về

Kết quả được in ra là: 1310

Nếu x, y được truyền theo giá trị - trả về, có nghĩa x, y chỉ là bản sao của giá trị thực của các tham số được truyền vào, nên nếu có sửa đổi giá trị tham số, thì cũng thay đổi giá trị thực của tham số đó. Nhưng khi kết thúc chương trình con, giá trị của x, y sẽ được gán trở lại cho các tham số thực.

x,y truyền theo tham chiếu:

Kết quả được in ra là: 1310

Do x, y được truyền theo tham chiếu nên x = \*i và y = \*a[i], do i =0 lúc truyền vào nên i = \*a[0], khi thực hiện x++ tương ứng i++ nên tại lúc đó i = 1 và y++ tương ứng a[0]++ nên a[0] = 3, do đó khi kết thúc hàm i = 1 và a = {3,1,0}

x,y truyền theo tên

Kết quả được in ra là: 1220

Do truyền theo tên nên x = i và y = a[i]. Khi thực hiện x++ tương đương i ++, khi đó i = 1. Khi thực hiện y++ tương ứng a[i] ++, do i lúc đó là 1 nên tương ứng a[1] ++, khi đó a[1] = 2. Khi kết thúc hàm i = 1 và a = {2,2,0}

=========================================================

Phan Thành Tín : 1915520

Trần Quyết Thắng: 1915251

Nguyễn Văn Khoa - 1812660

Question 1:

int mul (Node root) {

if (root == null) retrun 1;

if (root.val == 0) return 1;

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

}

Question 2:

Truyền bằng giá trị

trong hàm increase (0, 2)

x++ -> x= 1, y++ -> y=3

Giá trị x và a[i] không đổi vì truyền bằng giá trị

Kq: 0210

Truyền bằng giá trị - kết quả

trong hàm increase (0, 2)

x++ -> x= 1, y++ -> y=3

Giá trị x, y được trả về cho i và a[i] sau khi hàm increase kết thúc

Kq: 1310

Truyền bằng tham chiếu

trong hàm increase (0, 2)

x++ -> x= 1 =i, y++ -> y=3 =a[0]

Kq: 1310

Truyền bằng tên

======================================================

Lại Đức Anh Khoa - 1910265

Bùi Xuân Thông - 1915350

Huỳnh Quang Thuận - 2134144

**Bài 1:**

int mul(Node root) {

if (root == null) return 1;

else if (root.val == 0)  return 0;

else if (mul(root.left) == 0) return 0;

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

}

**Bài 2**:

Kết quả: 0 2 1 0

Bởi vì các tham số của hàm increase() được passed by value nên chỉ nhận vào các giá trị copy. Do đó các thay đổi trong thân hàm không làm thay đổi giá trị của biến ở ngoài

Kết quả: 1 3 1 0

Khi truyền bằng value-result, các tham số được truyền vào dưới dạng giá trị copy. Do đó, i tăng lên 1, a[0] tăng giá trị thành 3 trong thân hàm. Sau khi hàm kết thúc, các giá trị trên được truyền ra ngoài nên cũng làm thay đổi các giá trị bên ngoài hàm.

Kết quả: 1 3 1 0

Khi pass by reference, các tham số được nhận vào dưới dạng địa chỉ, do đó mọi thay đổi trong thân hàm chính là thay đổi của biến được truyền vào. Do đó, i tăng lên 1, a[0] tăng giá trị thành 3 khi hàm thực thi.

Kết quả: 1 2 2 0

Khi pass by name, các tham số được nhận vào đóng vai trò là biến ở bên ngoài, do đó mọi thay đổi trong thân hàm chính là thay đổi của biến được truyền vào. Do đó, i tăng lên 1, a[i] sẽ thành a[1] và tăng giá trị thành 2 khi hàm thực thi.

===========================================================

Member:

Trương Vĩnh Phước - 1910473

Nguyễn Văn Quang - 1914802

Hoàng Văn Hiếu - 1913328\

Lê Hoàng Anh - 1910752

Quest 1.

int mul (Node root){

if(root == null) return 1;

int lroot = mul(root.left);

int rroot = mul(root.right);

if(root.val == 0 || lroot ==0 || rroot == 0) return 0;

return root.val \* lroot \* rroot;

}

Quest 2.

x,y are passed by value

In ra 0210

Vì passed by value copy giá trị của i, a[i] (với i = 0)vào biến x,y, khi x,y thay đổi thì giá trị của i và a[i] không bị ảnh hưởng.

x,y are passed by value-result

In ra 1310

Vì với passed by value-result, giá trị của i, a[i] (với i = 0) được copy vào biến x,y.

x++ -> x = 1

y++ -> y = 3

Giá trị mới của x,y sau đó được copy lại cho i, a[i] (với i = 0)

      C. x,y passed by reference

In ra 1310

Với pass by reference, ta truyền vào tham khảo của biến, tức địa chỉ của biến đó vào hàm, từ đó mọi thay đổi diễn ra với giá trị các biến x, y thực tế diễn ra trực tiếp trên bộ nhớ của 2 biến đầu vào, từ đó thay đổi giá trị của chúng.

D. x,y passed by name

In ra 1220

Với passed by name, x,y là tên gọi khác đại diện cho i, a[i] (với i = 0)

trong hàm increase:

x++ -> i++ -> i = 1

y++ -> a[i] ++ -> a[1]++ -> a[1] = 2

=====

Nguyễn Kim Phương Trang - 1910620

Võ Văn Tiến Dũng - 1910984

Lê Cao Minh Trí - 1915651

Câu 1

int mul(Node root){

if (root == null) return 1;

else if (root.val == 0) return 0;

else if (root.left && root.left.val == 0) return 0;

else if (root.right && root.right.val == 0) return 0;

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

}

Câu 2

pass by value

i = 0 -> x = 0

a[i] = 2 -> y = 2

Vì x, y passed by value nên sự thay đổi của x, y trong hàm không thay đổi giá trị i, a[i] truyền vào,

In ra: 0210

pass by value-result

x, y nhận giá trị của i, a[i], tức là 0 và 2(a[0])

2 argument của hàm sẽ là i và a[0].

giá trị của x, y sau khi thực thi hàm sẽ được truyền ngược vào 2 argument là i, a[0]

x = 1 -> i = 1

y = 3 -> a[0] = 3

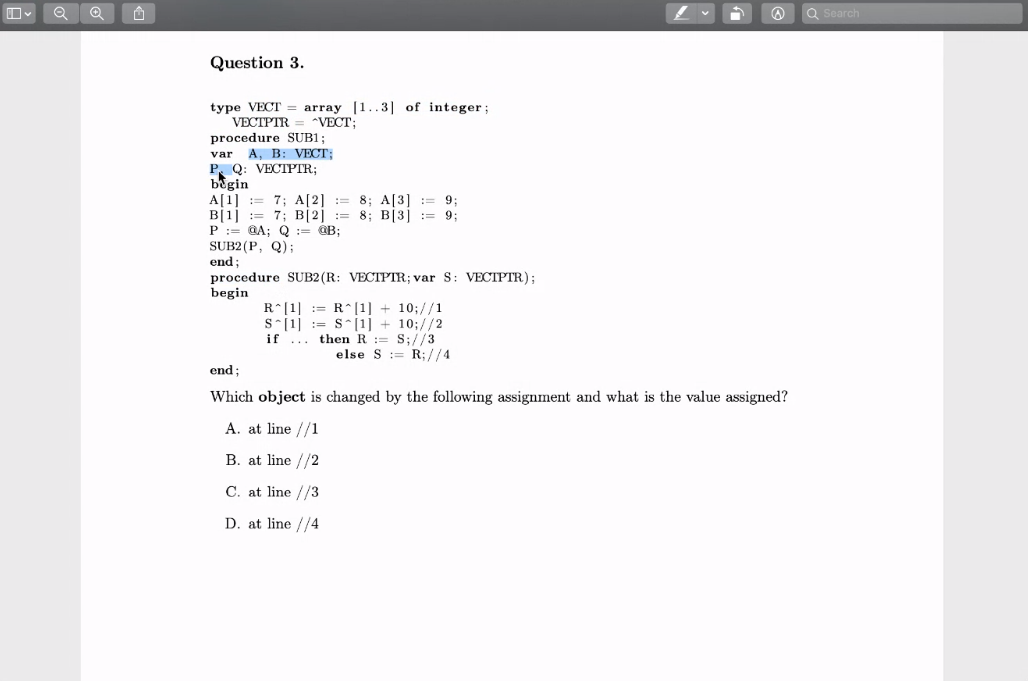
In ra: 1310

Pass by reference:

x, y nhận địa chỉ của i, a[0], giá trị x, y thay đổi cũng thay đổi trên i, a[i]

sau khi thực thi hàm: x = 1, y = 3 → i = 1, a[0] = 3

In ra 1310



====================================================================

Võ Nguyễn Thiện Nhân 1919409

Lê Tiến Dũng - 1811775

Nguyễn Văn Hoàn - 1711376

con trỏ P lưu địa chỉ ô nhớ đầu tiên của mảng A. Biến R được truyền tham trị (truyền P vào R) nên R cũng lưu địa chỉ của ô nhớ đầu tiên của mảng A

còn trỏ Q lưu địa chỉ ô nhớ đầu tiên của mảng B. Biến S được truyền tham khảo (truyền Q vào S) nên S lưu địa chỉ của biến Q

at line 1

object changed: A[1] → A[1] = 17

vì R cũng lưu giá trị của ô nhớ đầu tiên của mảng A nên khi R thay đổi cũng có nghĩa là mảng A bị thay đổi theo

at line 2

object changed: B[1] → B[1] = 17

vì S lưu địa chỉ của biến Q (và biến Q lưu địa chỉ ô nhớ đầu tiên của mảng B) nên khi S thay đổi thì mảng B bị thay đổi theo

at line 3

object changed: R

biến R sẽ bị thay đổi, lưu địa chỉ mảng B nhưng biến P không bị thay đổi vì biến P được truyền vào thông số hình thức theo kiểu tham trị nên nó sẽ copy giá trị của P gán cho R → P không bị thay đổi

at line 4

object changed: S và Q

biến S và Q bị thay đổi, lưu trữ địa chỉ của mảng A vì biến Q được truyền vào thông số hình thức biến S theo kiểu tham khảo cho nên mọi sự thay đổi của S đều ảnh hưởng đến Q

====================================================================

Nguyễn Hồng Thiện - 1915290

Trần Tiến Lộc - 1914038

Huỳnh Thành Đạt - 1910110

Question 3:

R truyền bằng value, S truyền bằng reference

Tại dòng //1

Giá trị A[1] thay đổi thành 17 vì:

Tính giá trị ô nhớ R[1], là ô nhớ A[1] = 7. Sau đó cộng thêm 10 → R[1] = A[1] = 17

Tại dòng //2

Giá trị B[1] thay đổi thành 17 vì

Tính giá trị ô nhớ S[1], là ô nhớ B[1] = 7. Sau đó cộng thêm 10 → S[1] = B[1] = 17

Tại dòng //3

R thay đổi thành S (Q, @B), nhưng P không đổi vì R truyền bằng value

Tại dòng //4

S đổi, Q đổi thành R (P, @A) vì S truyền bằng reference, S đổi dẫn đến Q thay đổi theo đổi thành địa chỉ con trỏ P (ô nhớ A)

========================================================================

Nguyễn Duy Kiên - 1812704

Nguyễn Hữu Anh Đại - 2020020

Lê Hồng Phong - 1813518

Q3.

Vì R được truyền bằng giá trị, nên R chứa địa chỉ của A.

R^[1] là giá trị ô nhớ thứ 1 trong mảng A

Nên dòng 1 ô nhớ thứ 1 mảng A tăng thêm 10. A[1] = 17

Object A[1] thay đổi

Vì S được truyền bằng tham chiếu, nên S == Q chứa địa chỉ của B

S^[1] là giá trị ô nhớ thứ 1 trong mảng B

Nên dòng 2 ô nhớ thứ 1 mảng B tăng thêm 10. B[1] = 17

Object B[1] thay đổi

R được gán bằng S, vì R được truyền bằng giá trị nên giá trị R = S, R chứa địa chỉ của B. Trong khi giá trị của P không đổi vẫn chứa địa chỉ của A. object R thay đổi

S được gán bằng R, vì S được truyền bằng tham chiếu nên phép gán S = R sẽ dẫn đến giá trị của Q = R, R chứa địa chỉ của A. object Q và bí danh S thay đổi

Nguyễn Kim Phương Trang - 1910620

Võ Văn Tiến Dũng - 1910984

Lê Cao Minh Trí - 1915651

Câu 3

R passed by value nên R được truyền giá trị của P là @A → A[1] thay đổi, giá trị mới là A[1] + 10 = 17

S passed by reference nên S được truyền địa chỉ của Q → S^[1] có địa chỉ của B[1] → B[1] thay đổi, giá trị mới là 7 + 10 = 17

R được pass by value → Object R thay đổi, giá trị mới là @B (vị @B được lưu trong S)

S được pass by reference → Object Q thay đổi, giá trị mới là @A (vì @A được lưu trong R)

Lại Đức Anh Khoa - 1910265

Bùi Xuân Thông - 1915350

Huỳnh Quang Thuận - 2134144

**Bài 3:**

R^[1]  chính là  A[1], khi đó A[1] = A[1] + 10 = 7 + 10 = 17

S^[1] ch B[1], khi đó S^[1] = S^[1] + 10 = 7 + 10 = 17 cũng chính là thay đổi trên B[1], do đó B[1] được thay đổi giá trị thành 17

Khi gán R:=S, vì R pass by value nên chỉ R thay đổi

Khi gán S:=R, vì S pass by reference nên S thay đổi, Q thay đổi trở thành địa chỉ của A

====================================================================

Phan Thành Tín : 1915520

Trần Quyết Thắng: 1915251

Nguyễn Văn Khoa: 1812660

Question 3:

param R pass by value, R copy giá trị của tham số truyền vào =>  R = P = @A.

Thay đổi R[1] tương ứng thay đổi giá trị của A[1] => A[1] = 7 + 10 = 17

param S pass by reference, S có cùng giá trị địa chỉ của tham số truyền vào => S tương đương Q = @B.

Thay đổi S[1] tương ứng thay đổi giá trị của B[1] => B[1] = 7 + 10 = 17

R là một biến mới chỉ tồn tại trong hàm sub2 gán S cho R chỉ thay đổi giá trị của biến R cục bộ. R = S = @B

S đại diện cho Q nên Q thay đổi: Q = R = P = @A

==============================================================

—-----------------------------------------------------------------------------------

Trương Vĩnh Phước - 1910473

Nguyễn Văn Quang - 1914802

Hoàng Văn Hiếu - 1913328

Lê Hoàng Anh - 1910752

Quest 3.

**A. // line 1:**

P được truyền (by value) vào R, và P giữ địa chỉ của A, nên R giữ địa chỉ của A, khi thay đổi giá trị  R^[1] thì giá trị của A[1] cũng thay đổi. Vậy có object A bị thay đổi. Giá trị mới của A: {17,8,9}

**B. // line 2:**

Q được truyền(by reference) vào S, và Q giữ địa chỉ của B, nên S giữ địa chỉ của B. S^[1] thay đổi nên  B[1] cũng thay đổi. B[1] = S^[1] = 17. Vậy có object B bị thay đổi.Giá trị mới của B: {17,8,9}

**C. // line 3:**

Giá trị của P được truyền vào R theo kiểu pass by value, nên khi R thay đổi thì P không thay đổi. Vậy chỉ có object R bị thay đổi, giá trị mới của R là địa chỉ của B.

**D.// line 4:**

Giá trị của Q được truyền vào S theo kiểu pass by reference, nên khi S thay đổi thì Q thay đổi. Trong trường hợp này giá trị của 2 object Q và S thay đổi, với giá trị mới là địa chỉ của A.

Nguyễn Ngô Thanh Trúc - 1910650

Nguyễn Minh Phú - 1914659

Lê Thanh An - 1912523

A thay đổi. Do hàm truyền vào P, con trỏ trỏ tới mảng A, giá trị truyền vào là địa chỉ ô nhớ A[1] nên ô nhớ A[1] sẽ bị thay đổi thành A[1] = 17

B thay đổi. Do hàm truyền vào Q, con trỏ trỏ tới mảng B, giá trị truyền vào là địa chỉ ô nhớ B[1] nên ô nhớ B[1] sẽ bị thay đổi thành B[1] = 17

P thay đổi. Do R được gán bằng S nên nó sẽ trỏ vào ô nhớ của con trỏ S

Q thay đổi. Do S được gán bằng R

=========================================================

Vũ Minh Dương: 1810885

Nguyễn Thọ Nam - 1911650

Lê Ngọc Minh Nhân - 1910402

Line 1: Giá trị A[1] bị thay đổi thành 17

Tính giá trị của R^[1], tức là A[1], là 7

Cộng giá trị đó cho 10, được 17

Lưu giá trị 17 vào cho R^[1], tức là A[1]. Suy ra: A[1] = 17

Line 2: Tương tự, B[1] thay đổi giá trị thành 17

Line 3: Không có thay đổi giá trị của A,B do R (lưu địa chỉ ô nhớ đầu của A) và S (lưu địa chỉ ô nhớ đầu của B) là biến cục bộ của SUB2; R = S tức R chuyển sang lưu địa chỉ ô nhớ đầu của B

Line 4: Không có thay đổi giá trị của A,B do R (lưu địa chỉ ô nhớ đầu của A) và S (lưu địa chỉ ô nhớ đầu của B) là biến cục bộ của SUB2; S = R tức S chuyển sang lưu địa chỉ ô nhớ đầu của A

