24/24 Questions Answered
Saved at 4:09 PM

TIME REMAINING
17 mins

FinalExam - Group 2

Q1 Subprogram Passing
12 Points

Given the following program

```
function Main()
int x = 8, y = 7, z= 0;
    function A();
    int x = 3, z= 1;
        function P()
        begin
            print (x, " ",y, " ", z);
        end;
        function B(y)
        int x = 5, w=20;
        begin
            C(P);
        end;
        function C(F);
        int y = 2, z = 10;
        begin
            F;
        end;
    begin
        y = 9;
        B(x);
    end.
begin
    A();
end.
```

Q1.1
4 Points

What will be the output if ad hoc binding is used, assuming that the language is statically scoped?

5 7 1

Q1.2
4 Points

What will be the output if shallow binding is used, assuming language is dynamically scoped?

5 2 10

Q1.3
4 Points

What will be the output deep binding is used, assuming the language is statically scoped?

3 7 1

Q2 Subprogram Implementation
12 Points

```
Program main
     procedure Bigsub is
     var x,y,z;
          procedure A(var x) is
          var u,z;

               procedure B is
               var z,w;
               begin   -- B
               y = x + u;  <-------------------- Point 1
               end; -- B

               procedure C is
               var x,u;
               begin -- C
               u = 5;
               x = 6;
               B();
               end; -- C
          begin -- A
          u = 3;
          v = 4;
          C();
          end; -- A
     begin -- Bigsub
     x = 1; y = 2; z = 3;
     A(x,z);
     end; -- Bigsub
begin -- main
    Bigsub()
end -- main
```

Given the above code, answer the questions below. Make sure that the local offsets you give are consistent with the activation record format used and are relative to the bottom of each activation record.

Q2.1

4 Points

Assuming that static scoping is used, what are the (chain_offset, local_offset) pairs for variable y at Point 1?

Chain offset

2

Local offset

3

Q2.2

4 Points

Assuming that static scoping is used, what are the (chain_offset, local_offset) pairs for variable x at Point 1?

Chain offset

1

Local offset

2

Q2.3

4 Points

Assuming that static scoping is used, what are the (chain_offset, local_offset) pairs for variable u at Point 1?

Chain offset

1

Local offset

3

Q3 Parameter Passing

15 Points

Given the following C-like code, answer the questions below.

```
int a,b;
int c[3] = {3, 2, 1};

void foo(int x, int y) {
   c[x-2] = 4;
   a--;
   if (x>2) {
       x--;
       foo(x, y);
       c[x-1] = 5;
       return;
   }
   b++;
   y++'
   c[2] = b+y;
   x = 3;
}

int main() {
   a = 3; b = 3;
   foo(a, b);
   printf("%d %d %d %d %d", a, b, c[0], c[1], c[2]);
   return 0;
}
```

Q3.1

2 Points

Write down the values that will be printed for the variables below if the parameters are passed by value.

a:

1

b:

4

Q3.2

3 Points

Write down the values that will be printed for the variables below if the parameters are passed by value.

c[0]:

4

c[1]:

5

c[2]:

8

Q3.3

2 Points

Write down the values that will be printed for the variables below if the parameters are passed by reference.

a:

3

b:

5

Q3.4

3 Points

Write down the values that will be printed for the variables below if the parameters are passed by reference.

c[0]:

3

c[1]:

4

c[2]:

10

Q3.5

2 Points

Write down the values that will be printed for the variables below if the parameters are passed by value-result (address computed at the return).

a:

3

b:

4

Q3.6

3 Points

Write down the values that will be printed for the variables below if the parameters are passed by value-result (address computed at the return).

c[0]:

4

c[1]:

4

c[2]:

5

Q4 Scheme

4 Points

Given the following Scheme functions, select whether they are tail recursive or not.

Q4.1

1 Point

```
(define (fun n total)
    (cond ((= n 0) 0)
          ((even? n) (fun (- n 1) total))
          (else (fun (- n 2) (+ total n)))))
```

- (●) tail-recursive
- ( ) not tail-recursive

Q4.2

1 Point

```
(define (fun x y)
    (if (or (= x 0) (= y 0) (= x y))
        1
        (if (< x y)
            (fun x (- y x))
            (if (> x y)
                (fun (- x y) y)))))
```

- (●) tail-recursive
- ( ) not tail-recursive

Q4.3
1 Point

```
(define (fun x)
  (if (or (= x 0) (= x 1))
   x
      (+ (fun (- x 1)) (fun (- x 2)))))
```

- ○ tail-recursive
- ◉ not tail-recursive

Q4.4
1 Point

```
(define (fun n curr next)
    (if (= n 0)
        curr
        (fun (- n 1) next (+ curr next))))
```

- ◉ tail-recursive
- ○ not tail-recursive

Q5 Static Scoping
10 Points

```
program MAIN
  var a, b, c;
  procedure sub1
    var b, c, d;
    procedure sub2(var e)
      begin {sub2}
        e = a + c + d;      /* Point A */
        print(a, b, c, d, e);
      end {sub2}
    begin {sub1}
      b = 2 * a;
      c = b - 1;
      d = a + b;
      sub3;
    end   {sub1}
  procedure sub3
    var a, c;
    procedure sub4(var d)
      begin {sub4}
        a = d + b ;
        sub2(a);
      end   {sub4}
    begin {sub3}
      a = 3;
      b = b + 1
      c = a + b;
      sub4(c);
    end   {sub3}
  begin {MAIN}
    a = 2;
    b = 5;
    c = 4
    sub1;
  end   {MAIN}
```

Answer below questions for the given code that uses static scoping.

Q5.1 Static Scoping

1 Point

Check the correct condition about the variables below at point A if the programming language uses static scoping. The variables are written as <var_name>(<declared_subprogram>).

a(MAIN)

- (●) visible
- ( ) hidden
- ( ) not available

b(MAIN)

- ( ) visible
- (●) hidden
- ( ) not available

c(MAIN)

- ○ visible
- ◉ hidden
- ○ not available

Q5.2 Static Scoping

2 Points

Select the correct condition about the variables below at point A if the programming language uses static scoping. The variables are written as <var_name>(<declared_subprogram>).

b (sub1)

- ◉ visible
- ○ hidden
- ○ not available

c (sub1)

- ◉ visible
- ○ hidden
- ○ not available

d (sub1)

- ◉ visible
- ○ hidden
- ○ not available

Q5.3 Static Scoping

2 Points

Select the correct condition about the variables below at point A if the programming language uses static scoping. The variables are written as <var_name>(<declared_subprogram>).

e (sub2)

- (•) visible
- ( ) hidden
- ( ) not available

a (sub3)

- ( ) visible
- ( ) hidden
- (•) not available

c (sub3)

- ( ) visible
- ( ) hidden
- (•) not available

d (sub4)

- ( ) visible
- ( ) hidden
- (•) not available

Q5.4 Static Scoping

5 Points

What will the output of the code be if the language uses static scoping?

a:

2

b:

4

c:

3

d:

6

e:

11

Q6 Dynamic scoping

10 Points

```
program MAIN
  var a, b, c;
  procedure sub1
    var b, c, d;
    procedure sub2(var e)
      begin {sub2}
        e = a + c + d;      /* Point A */
        print(a, b, c, d, e);
      end {sub2}
    begin {sub1}
      b = 2 * a;
      c = b - 1;
      d = a + b;
      sub3;
    end   {sub1}
  procedure sub3
    var a, c;
    procedure sub4(var d)
      begin {sub4}
        a = d + b ;
        sub2(a);
      end   {sub4}
    begin {sub3}
      a = 3;
      b = b + 1
      c = a + b;
      sub4(c);
    end   {sub3}
  begin {MAIN}
    a = 2;
    b = 5;
    c = 4
    sub1;
  end   {MAIN}
```

Answer below questions for the given code that uses dynamic scoping.

Q6.1 Dynamic scoping

1 Point

Select the correct condition about the variables below at point A if the programming language uses dynamic scoping. The variables

are written as <var_name>(<declared_subprogram>).

a(MAIN)

- ○ visible
- ◉ hidden
- ○ not available

b(MAIN)

- ○ visible
- ◉ hidden
- ○ not available

c(MAIN)

- ○ visible
- ◉ hidden
- ○ not available

Q6.2 Dynamic Scoping

2 Points

Select the correct condition about the variables below at point A if the programming language uses dynamic scoping. The variables are written as <var_name>(<declared_subprogram>).

b (sub1)

- ◉ visible
- ○ hidden
- ○ not available

c (sub1)

- ○ visible
- ◉ hidden
- ○ not available

d (sub1)

- ○ visible
- ◉ hidden
- ○ not available

Q6.3 Dynamic Scoping

2 Points

Select the correct condition about the variables below at point A if the programming language uses dynamic scoping. The variables are written as <var_name>(<declared_subprogram>).

e (sub2)

- ◉ visible
- ○ hidden
- ○ not available

a (sub3)

- ◉ visible
- ○ hidden
- ○ not available

c (sub3)

- ◉ visible
- ○ hidden
- ○ not available

d (sub4)

- ◉ visible
- ○ hidden
- ○ not available

Q6.4 Dynamic Scoping

5 Points

What will the output of the code be if the language uses dynamic scoping?

a:

13

b:

5

c:

8

d:

8

e:

29

Submit & View Submission