COMP 141: Control

Instructions: In this exercise, we are going to review environments with activation records.

(1) C is a statically scoped PL. Consider the following C program.

```
#include <stdio.h>

int a = 1;
int b = 2;
int c = 0;


int g (int a) {
        int c = b;
        int b = a + c;
        printf("5: %d %d %d\n", a, b, c);
        return b;
}


int f (int a){
        int b = a + 5;
        printf("2: %d %d %d\n", a, b, c);
        {
                int a = 4;
                b = a + 2;
                printf("3: %d %d %d\n", a, b, c);
        }
        printf("4: %d %d %d\n", a, b, c);
        return g (b);
}

int main () {
        int c = 3;
        printf("1: %d %d %d\n", a, b, c);
        int a = f (c);
        return 0;
}
```

(a) Define the environment in each of the points in the program that a `printf` exists.

(b) Specify the defining and calling environments for each of the procedure and non-procedure blocks. Note that for non-procedure blocks the defining and calling environment are the same as the surrounding block.

(c) According to the environment in each of the points in the program that a `printf` exists, specify the output of this program.

(2) Consider the following code excerpt in a given PL.

```
global: x, y, z;

f(a,b,c) {
        a := 3;
```

```
        b := 4;
        c := 5;
        return a+b+c;
}

main() {
        x := 0;
        y := 1;
        z := 2;
        t := f(x,y,z);
        print(x,y,z);
}
```

What would be the result of printing if

(a) pass by value is used,

(b) pass by reference is used,

(c) pass by value-result is used.

(3) Consider the following code excerpt in a given PL.

```
global x,y;

f(a,b,c) {
        a := a + 1;
        b := b + 2;
        c := c + 3;
        return a + b + c;
}

main() {
        x := 0;
        y := f(x, x, x);
        print (x, y);
}
```

What would be the result of printing if

(a) pass by value is used,

(b) pass by reference is used,

(c) pass by value-result is used, where last function parameter is copied lastly.

(d) pass by value-result is used, where first function parameter is copied lastly.

(4) Consider the following code excerpt in a given PL.

```
global x,y,z;

f(a,b,c) {
        x := x + 1;
        y := y + 2;
        z := z + 3;
        return a + b + c;
}
```

```
main() {
        x := 0;
        y := 0;
        z := 0;
        t := f(x+y, y+z, x+z);
        print (x, y, z, t);
}
```

What would be the result of printing if

(a) pass by value is used, and

(b) pass by name is used.

(5) Consider the following code excerpt in a given PL.

```
global x,y,z;

f(a,b,c) {
        x := a;
        y := b;
        z := c;
        return a + b + c;
}

main() {
        x := 0;
        y := 1;
        z := 2;
        t := f(x+y, y+z, x+z);
        print (x, y, z, t);
}
```

What would be the result of printing if

(a) pass by value is used, and

(b) pass by name is used.

(6) Consider the following code excerpt in a given PL with pass by name semantics.

```
global: x, y, z;

f(a,b,c) {
        a := 3;
        b := 4;
        c := 5;
        return a+b+c;
}

main() {
        x := 0;
        y := 1;
        z := 2;
        local t := f(x,y,z);
        return;
```

```
}
```

Define the stack of activation records and environment pointer in the following points:

(a) when `main` is invoked

(b) when `f` is invoked

(c) before returning from `f`

(d) when `f` is returned

(e) when `main` is returned

(7) Answer the previous question, if pass by reference is used.

(8) Answer the previous question, by extending activation records to include access links.

(9) Consider the following code excerpt in a given PL with pass by name semantics.

```
global x;

f(a) {
        if a<=1 then return 1;
        else return a * f(a-1);
}

main() {
        x = 2;
        y = f(x);
        return;
}
```

Define the stack of activation records and environment pointer when the stack is in its largest size.

(10) Consider the following code excerpt in a hypothetical PL that initially runs `main()` function (Similar to C-like PLs).

```
global x,y,z;

g(a,b,c) {
        a := a * x;
        b := b - y;
        c := c * x;
        local d := a + b + c;
        return d;
}

f(a,b,c) {
        local d := a+b;
        if (d < 10) then return g(a,b,c);
        else return 5;
}

main() {
        x := 4;
        y := 3;
        z := 2;
```

```
        local w := f(x,y,z);
        return;
}
```

Define the stack of activation records and environment pointer i) before `g` returns, and ii) before `main` returns, in the following parameter passing styles.

(a) pass by value in both `f` and `g`

(b) pass by reference in both `f` and `g`

(c) pass by value result in both `f` and `g`

Note that the activation records must include the following details:

- control link
- access link
- return address
- function parameters
- local variables
- temporaries