CS383 Assignment 5

Instructions:

- *Submit all your answers in hard copies.*
- *Please submit source codes for questions 1, 3 and 9 to [alex_wang@126.com.](mailto:alex_wang@126.com)*
- *This assignment is released on 12/4/2011, and due 12/8/2011.*
- *Please remember to include your name and student ID on all copies.*

1. For the following Clite program,

```
int Fibonacci (int n) {
   if (n < 2) return n;
   else return Fibonacci(n-1) + Fibonacci(n-2);
}
int main () {
 int answer;
 answer = Fibonacci(8);
}
```

(a) What is the value of the parameter n in topmost stack activation record each time the function Fibonacci calls?

(b) How many stack activation records are activated for the call Fibonacci(13)?

(c) Think of a different way to define the function Fibonacci so fewer stack activation records are activated. Write down your code and answer question (b) on your implementation.

2. The following C/C++ program solves the Towers of Hanoi problem for three disks.

```
void MoveTower(int disks, char start, char end, char temp)
{
     if(disks ==1)
          cout<<"Move a disk from "<<start<<" to "<<end<<endl;
     else{
          MoveTower(disks-1,start,temp,end);
          cout<<"Move a disk from "<<start<<" to "<<end<<endl;
          MoveTower(disks-1,temp,end,start);
     }
}
int main(int argc, char* argv[]) {
     int totalDisks = 3;
     MoveTower(totalDisks,'A', 'B', 'C');
     return 0;}
```

Draw the Run-Time Stack after each call and return of MoveTower function.

3. How can the Ada procedure in Figure 9.9 (page 240) be rewritten in C/C++/JAVA, which does not allow nesting of functions? Design 4 tests to test your implementation.

4. Run the following Clite program, and draw the Abstract Syntax Sketch (like Figure 10.4) for this program.

```
int rem (int x, int y){
  return x - x/y * y;
}
int gcd (int x, int y){
    int z;
    if (y == 0) return x;
    else if (x == 0) return y;
    else {
     z = rem(x, y);
 return gcd(y, z);}}
int main () {
    int answer;
    answer = gcd(24,10);}
```

5. Write down type map tm_G, tm_F and tm_f for the following program, where f can be either main, A, B or C.

```
int h, i, j,k;
void C(int m, int l, int n){
        h = m + l + n;}
void B(int w) {
        int j, k;
        i = 2*w;
        w = w + 1;
        C(i, j, w);}
void A(int x, int y) {
        bool i, j;
        B(h);}
int main () {
        int a, b;
        h = 5; a = 3; b = 2; k = 12;
        A(a, b);}
```

6. Using the example Clite functions defined in Figures 10.1 and 10.5, construct three different calls; each call should violate one of the Type Rules 10.6, 10.7, and 10.8, but not the other two.

7. The state resulting from executing a *Call c* to a void function *f* in a Clite Program is defined as below:

$$
\begin{aligned}
&M : \text{Call} \times \text{Function} \times \text{State} \rightarrow \text{State} \\
&M(c, f, \sigma) = removeactivationrecord(f.params, f.locals, \\
&\quad M(f.body, ByValue(f.params, c.args, \\
&\quad\quad addactivationrecord(f.locals, f.params, \sigma))))
\end{aligned}
$$

Define mathematically function $removeactivationrecord$ and function $addactivationrecord$.

8. Read the following Perl program, get it to run, and explain why some of addresses printed are identical.

```
#!/usr/local/bin/perl
Use strict;
Use warnings;
my @array;
for(0..9){
    my $tmp = 123;
    my $addr = \$tmp;
    Print "$_ has address: $addr\n";
    $array[$_/2] = $addr;
}
```

9. Exercise 11.9 on book, page 276.