

CSE 305 Spring 2015
Assignment #3: Recursion, Scope Rules, and Binding Time

Assigned: Mar 10, 2015
Due: Mon, Mar 30, 2015 (11:59 pm)

Note: This assignment may be done by a pair of students.

1. This question is about using static variables to define a history sensitive behavior. Write a C function called `fibonacci` whose behavior is as follows: When `fibonacci(i)` is called for the i^{th} time, it returns the i^{th} number in the Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21, ... Thus, the 1st call on `fibonacci()` returns 1; the 2nd call on `fibonacci()` returns 1; the 3rd call returns 2; the 4th call returns 3; etc.

Important: The `fibonacci()` definition should not take any parameters or refer to global variables; nor should it use loops (for-loops, while-loops, etc.) nor call other functions. History-sensitive behavior means that `fibonacci()` should compute each new number as the sum of the previous two numbers in the series. The outline of `fibonacci()` is as follows:

```
int fibonacci() {  
    // declare one or more static variables  
    // a series of assignment statements; if-else is permitted, if you wish  
    // finally, return the answer  
}
```

2. This question is about static scoping, recursion, and higher-order procedures in the C programming language. Run the following C program:

```
#include <stdio.h>  
int main() {  
    int x, y;  
    void p1(int y, void q(int)) {  
        void p2(int x) {  
            x = y + 2;  
            printf("%d\n", x);  
            q(y);  
        }  
        if (x == y)  
            q(y);  
        else p1(y+1, p2);  
    }  
    void p2(int x) {  
        x = y + 2;  
        printf("%d\n", x);  
    }  
    x = 2; y = 2;  
    p1(0, p2);  
}
```

- What is the sequence of values printed?
- Draw a Scope Diagram at the point when the last value was printed.
- Draw a Stack Diagram at the point when the last value was printed.

Notes:

- In the Scope Diagram, for every call on p1, be sure to show the “closure” that is passed to q in addition to the value bound to y. The closure is a pair <procedure-name, environment >. it is important to nest the frames for the procedure calls correctly.
 - In the Stack Diagram, suffices to show the names of the stack frames along with the static and dynamic links. Details of the parameters for the procedures p1 and p2 are not required.
3. This question is about heap-dynamic binding. Refer to the program bfirst.c posted on Piazza: Resources→Homeworks. This program does a bread-first traversal of a tree. The heart of the program is the void procedure bfirst shown below.

```
void bfirst(TREE *tr, TREELIST* f(TREELIST*, TREE*)) {
    void bf(TREELIST *tl) {
        TREE* tr;
        while(tl != NULL) {
            tr = tl->tree;
            printf("%d ", tr->value);
            tl = tl->next;
            if (tr->left != NULL) tl = f(tl, tr->left);
            if (tr->right != NULL) tl = f(tl, tr->right);
        }
    }
    bf(cons(tr, NULL));
}
```

The procedure bfirst is parameterized by the tree to be traversed as well as a procedure f. The parameter f receives a procedure that adds a tree at the back of a list of trees. There are two different procedures that can be used for this purpose:

- enqueue(tlist, tr), which adds tr to tlist by updating the tail of tlist; and
- append(tlist, tr), which returns a new list with the tree tr after all elements in tlist.

a. The procedure bfirst will have different heap storage needs depending upon whether enqueue or append is passed to it. For each of these two cases, complete the table below by showing the number of cons cells allocated on the heap during each iteration of the while-loop in the procedure bf. The tree being traversed is the same in both cases, and it is given in file bfirst.c.

Iteration #	# of cons cells allocated
1	...
2	...
...	...
n	...

(b) For each of the two tables in part (a), what is the total number of garbage cells on the heap when bfirst is about to complete its execution? A garbage cell is one that cannot be accessed from the stack frame of main().

Online Submission:

1. This assignment may be done by a team of at most two students. Write both student names at the top of the file.
2. No handwritten submissions permitted; all answers must be prepared using a word processor.
3. Name the file A3.pdf. Both students should submit the file A3.pdf from their respective CSE accounts.
4. It is fine to do the assignment solo. Write your name at the top of the file and submit it.

End of Assignment #3