Computer Science Foundation Exam

August 1, 2003

COMPUTER SCIENCE I

Section I B

No Calculators!

KEY

Name: ______________________________

SSN: ______________________________

Score: 50

In this section of the exam, there are three (3) problems

You must do all of them.

The weight of each problem in this section is indicated with the problem. The algorithms in this exam are written in C. Any algorithms that you are asked to produce should use a syntax that is clear and unambiguous. Partial credit cannot be given unless all work is shown.

As always, be complete, yet concise, and above all <u>be neat</u>. Credit cannot be given when your results are unreadable.

1. **(15 points)**

   Write a **recursive** function that will concatenate two singly-linked lists. Assume that each list has already been created and may or may not be empty. The function is to be implemented requiring only two parameters, which may be assumed to be pointers to the lists to be concatenated.

   You may assume the following functions exist:

   ListHead (list) - this function returns a pointer to the head of the argument list.
   ListTail (list) - this function returns a pointer to the remainder of the argument list after removing the head of the list.
   ListCons(head, tail) –this function allocates and returns a new list, which consists of the element head followed by the list tail.

<u>One Possible Solution</u>

```
listADT  ListConcat (listADT list1,   listADT list2)
{
          if (list1 == NULL)
                  return (list2);
          else
                 return (ListCons(ListHead(list1), ListConcat(ListTail(list1), list2)));
}
```

2. (15 points – 5 points each)

Find the closed form expression in terms of the parameter N or an exact value if the summation limits are known, for each of the following summations. Show all of your work; an answer alone is not sufficient to receive full credit.

a) $\sum_{i=0}^{N}(5i+2) =$

Solution:

$$\sum_{i=0}^{N}(5i+2) = 5\sum_{i=0}^{N} i + 2\sum_{i=0}^{N} 1 = \frac{5(N)(N+1)}{2} + 2N + 2 = \frac{5N^2+9N+4}{2}$$

---

b) $\sum_{i=1}^{2N-4}(4i+2) =$

Solution:

$$\sum_{i=1}^{2N-4}(4i+2) = 4\sum_{i=1}^{2N-4} i + 2\sum_{i=1}^{2N-4} 1 = \frac{4(2N-4)(2N-3)}{2} + 2(2N-4)$$

$$= \frac{4\left(4N^2-6N-8N+12\right)}{2} + 4N - 8 = 2(4N^2-14N+12) + 4N - 8$$

$$= 8N^2 - 28N + 24 + 4N - 8 = 8N^2 - 24N + 16$$

---

c) $\sum_{i=14}^{27}(3i-2) =$

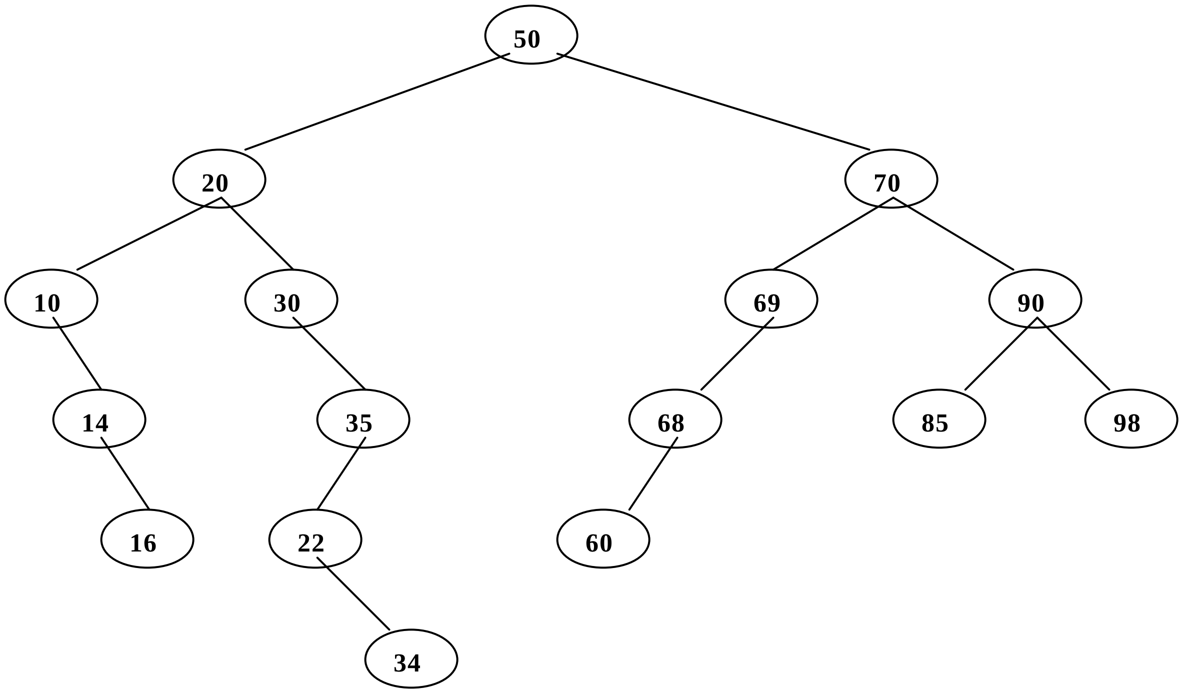
Solution:

$$\sum_{i=14}^{27}(3i-2) = \left(3\sum_{i=1}^{27} i - 3\sum_{i=1}^{13} i\right) - \left(2\sum_{i=1}^{27} 1 - 2\sum_{i=1}^{13} 1\right) = \frac{3(27)(28)}{2} - \frac{3(13)(14)}{2} - 2(27) + 2(13)$$

$$= 3(27)(14) - 3(13)(7) - 2(27) + 2(13) = 1134 - 273 - 54 + 26 = 833$$

3. (20 points – 2pts(a), 6pts(b), 12pts(c))

Given the following Binary Tree, answer the questions (a) through (c) below :



(a) Is this a valid Binary Search Tree? (circle one) YES NO

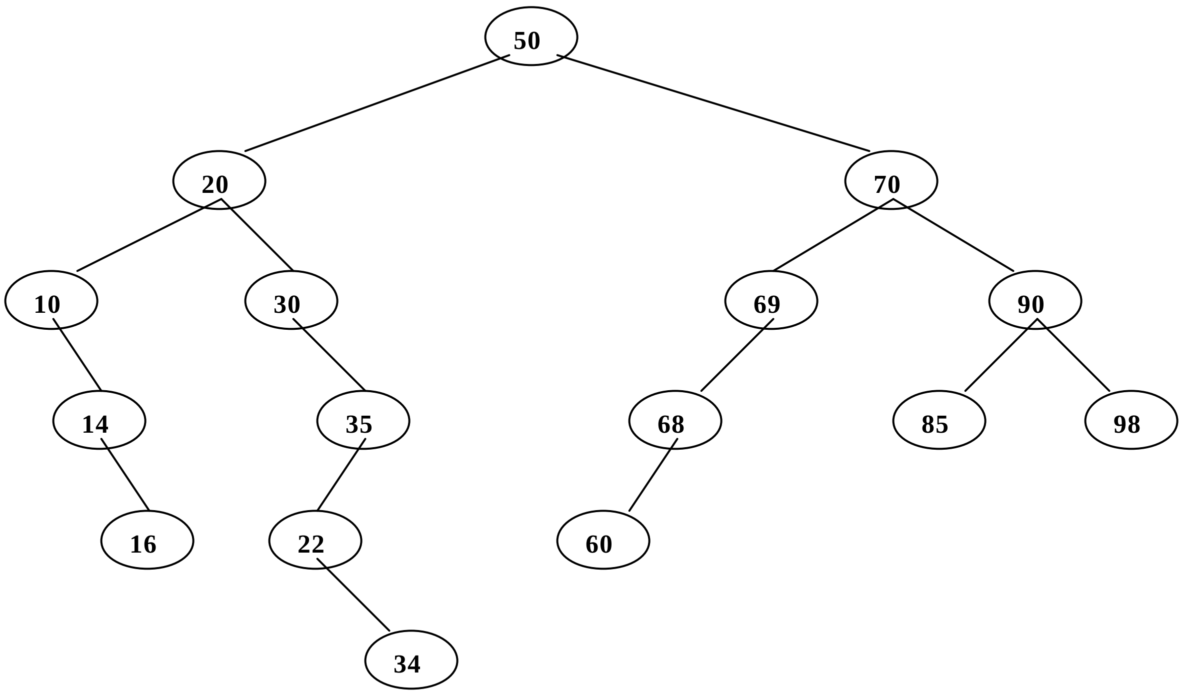
(b) List the nodes of this tree in the order that they are visited in a postorder traversal:

50	20	10	14	16	30	35	22

1st node visited

34	70	69	68	60	90	85	98

last node visited



(c) Execute the algorithm shown below using the tree shown above. Show the exact output produced by the algorithm. Assume that the initial call is: **prob3(root)** and that the tree nodes and pointers are defined as shown.

```
struct treeNode{
     int data;
        struct treeNode *left, *right;
}
struct treeNode *tree_ptr;

void prob3(struct tree_ptr *node_ptr) {
   if (node_ptr != NULL){
        if (node_ptr->data % 3 == 0){
             printf("%d ", node_ptr->data);
             prob3(node_ptr->left);
        }
        else{
             if (node_ptr->data % 3 == 1){
                  printf("%d ", node_ptr->data+2);
                  prob3(node_ptr->right);
             }
             else{
```

```
                if (node_ptr->data % 3 == 2){
                    prob3(node_ptr->left);
                    prob3(node_ptr->right);
                }
            }
        }
    }
}
```

Output:

12	**18**	**30**	**72**	**90**	**87**					