Indian Statistical Institute
Computer Science I (Programming in C)
First Year Students
30 November 2012

Maximum time: Three hours

Answer Any FIVE questions. Total Marks: 75

Question 1. [5+5+5]

a. Explain the term Storage Class Specifiers in C and their usage (scope and duration) using the storage class specifiers auto, extern and static. In a typical declaration of a variable such as " int i=1; '' what is the storage class spciifier of the variable i ?

b.)What are the outputs of the following programs?

Program 1

```
#include <stdio.h>
void stat();
int main()
{
  int i;
  for(i =1; i<=3; i++)
     stat();
 }
void stat()
{
  static int x=0;
  x = x+1;
printf("x=%d\n",x);
}
```

Program 2

```
#include <stdio.h>
void stat();
int main()
{
  int i;
  for(i =1; i<=3; i++)
     stat();
 }
void stat()
{
  int x=0;
  x = x+1;
printf("x=%d\n",x);
```

Question 2. [2+3+3+7]

Let a polynomial P(x) of x with integer coefficients be represented by a linked list as follows:

Each node of the linked list represents a non zero term in the polynomial . Each node stores the power of x , and its (nonzero) coefficient. Each node points to the next nonzero term in the polynomial.. For example the polynomial $2x^4+5x+7$ is represented by 4, 2 → 1, 5→ 0, 7 →NULL with obvious notation. Now do the following:

a.) Define a C structure for this linked list as well as pointer to it. (Use the above example as guide)
b.) Write a C function that accepts a pointer to a linked list representing a polynomial and returns the degree of the polynomial (Use the convention that power of x in each node is greater than the power of x in the succeeding node as in the example above)
c.) Write a C function that creates the linked list representing the polynomial 1.The function must return a pointer to the linked list.
d.) Write a C function that takes an integer n and a pointer to a linked list representing a P(x) and returns a pointer to a linked list that represents the product of x^n and P(x)

Question 3.[5+5+5]

Derive the recursion relations for the computational complexity T(n) in the following cases:

1. Tower of Hanoi problem: Moving n rings from one peg to another using a third peg by using the solution of moving n-1 rings, and then using recursion.

2. Finding the largest integer in an array of n integers by using the fact Max(a,b,c, ….) = Max(a, Max(b,c,…..)), and using recursion.

3. Solve the recursion equation in any one of the above case and give asymptotic estimate of T(n) for large n in that case.

Question 4. [2x5+5]

a. Explain the process of merge sort and heap sort by sorting the following sequence of integers 5,3,2,6,1,8,4,3. Show all the steps involved. Your discussion must be general enough to explain these processes for any such sequence and not just the above example.

b. Compare these two sorting processes from the point of view of time complexity of computation and use of memory space during the process. Which advantage, if any, does merge sort have over merge sort and vice versa.

Question 5. [4+ 7+4]

a. What is a stack? What are the basic operations defined on a stack?

b. Using the following implementation of a stack

```
typedef  struct node{
        int num;
        struct node *next;

}Node, *NodePtr;

Typedef struct {
NodePtr  top;
}StackType, *Stack;
```

define a C function that creates an empty stack and another C function to place a number on top of the stack

c. Using a stack provide an algorithm (not a C program) that takes an integer as input and gives the binary representation of the integer . Explain with an example.

Question 6.[4+6+5]

a. Create a binary search tree (b.s.t) for the following keys (numbers) assuming that they are inserted in the following order 56, 30, 61, 39, 47, 35, 75, 13, 21, 64, 26, 73, 18. (Please note 56 will be the root and as numbers are supplied you must ensure that it remains a b.s.t at all stages.)

b. Write the sequence of nodes in preorder (VLR) and inorder (LVR) traversals.

c. Using the result of b.) above, create the almost complete binary tree that represents the above sequence. Explain the process or algorithm you are using.