Indian Institute of Technology, Kharagpur

Department of Computer Science and Engineering

Class Test 1, Spring 2014
Programming and Data Structures (CS 11001)

Full marks: 40 | Date: 05-Feb-14 | Time: 1 hour

Name	Roll No.	Section	Marks Obtained

1. Answer ALL questions.

2. Answer all questions in the space provided in this question paper itself. Use the designated spaces and the last sheet for rough work.

3. Marks for every question are shown with the question.

---

1. Consider the following program

```
#include <stdio.h>
void main()
{
    int x, y, z, t;
    x = 45;  y = 35;  z = 40;
    printf("x=%d, y=%d, z=%d\n", x, y, z);
    if (x < y) { z += 2 * y;  y = 2 * x;  x = z; }
    printf("x=%d, y=%d, z=%d\n", x, y, z);
    if (x = z) { y = 2 * z;  z += 2 * x;  x += y; }
    printf("x=%d, y=%d, z=%d\n", x, y, z);
    if (y > z) { x = z;  z = 2 * y;  y = x; }
    else if (x < z) { y = 2 * z; z += x; x = 2 * y; }
    printf("x=%d, y=%d, z=%d\n", x, y, z);
}
```

(a). Write the output of the program [4]

(b). Write the output of the program if the line
x=45; y=35; z=40; is replaced by *x=25; y=35; z=45;* [4]

Ans (a)		Ans (b)

2. Write a complete C program to perform the following steps. Read in two positive integers n and k. Then read in *n* (> 0) positive integers. Output those numbers that are greater than the number *k* you have read. If there is no number greater than *k* print that such a number has not been found. DO NOT USE ARRAYs. Ensure that your program is able to detect wrong input that is if any of the numbers inputted is negative or zero, the program should exit. [10]

3. Consider the following program and answer the corresponding questions in the appropriate boxes after each question.

```
#include <stdio.h>

void main()
{
        int num, i, p, k;

        printf("\nInput an integer:");
        scanf("%d",&num);

        i = num;  p = 0;  k = 0;
        while (i > 0)
        {
        ____________________


        ____________________


        ____________________
        }

        printf("Reverse -- %d",p);

        _____________________


        _____________________
}
```

(a). Write the three lines inside the loop so that the value of p will be the reverse of the input value of *num*. For example, if num = 123, p = 321. You are not allowed to use any new variable. [3]

(b). Write down the next two lines which will check and declare (print) if the number *num* is a palindrome. Examples of palindrome numbers are 8, 22,121, 8778, 12321 etc. [2]

(c). Write the output of *p* if *p* is initialized to 1 (*p* =1) when *num* is 349. [1]

p =

(d). Write the output of *p* if *k* is initialized to 1 (*k* =1) when *num* is 675. [1]

p =

4. Consider the following program

```
#include <stdio.h>

void main()
{
        int j = _____, k = _________, i = __________;

        while (i < 10)
        {
          i++;
          j = j + i;
          k = k + j;

          printf("%d ",k);
        }
}
```

What should be the initial values of i, j, k if the output is [5 x 3]

(a). 4 11 23 41 66 99 141 193

i =	j =	k =

(b). 1 4 10 20 35 56 84 120 165 220

i =	j =	k =

(c). 7 15 27 44 67 97 135 182 239

i =	j =	k =