CS13002 Programming and Data Structures, Spring 2005

Class test 1

Total points: 30 | February 08, 2005 | Total time: 1 hour

Roll no: _____ **Name:** _____ **Section:** _____

Write your answers in the question paper itself. You may use extra blank sheets for rough work, but your answers must fit in the respective spaces provided. Answer all questions.

1. The following is the skeleton of a C program that computes the number of each numeral in a string. Fill in the blanks with appropriate C constructs. **(10)**

```
#include <stdio.h>

#define base 10

main ()  /* This program outputs the numbers of 0's,1's, ..., 9's in an
            input string ending in $ */
{
   char b;
   int i, a[base];

   /* Initialize array elements to zero */

   for ( _____ ; _____ ; _____ )
      a[i] = 0;

   printf("Input numeric characters ending with $\n" );

   scanf("%c", &b); /* Scan next character */

   /* Execute the loop as long as $ is not scanned */

   while ( _____ ) {
      printf("Processing the digit %c\n", b);

      /* Increment the count for the new digit */

      a[_____
      scanf("%c", &b); /* Scan next character */
   }

   for ( i=0 ; i<=9 ; i=i+1 )

      printf("There are %___ %___'s\n", a[i], i);
}
```

Note: You don't have to know the exact ASCII values for the characters $0, 1, 2, \ldots$. It is sufficient to know only that the ASCII representations of $0, 1, 2, \ldots$ are consecutive. Not all blanks carry equal marks. Evaluation will depend on the overall correctness.

2. Determine what the following program prints. **(10)**

```
#include <stdio.h>
main ()
{
    int r, t, m;
    scanf("%d", &r); /* Enter the last four digits of your roll number as r */
    printf("r = %d\n", r);
    m = 0;
    while (r > 0) {
        t = r % 10;
        if (t > m) m = t;
        r = r / 10;
    }
    printf("m = %d\n", m);
}
```

Write your answer in the space below this line:

```
r =

m =
```

Description in one sentence:

3. Complete the following program that scans a positive integer $n \geqslant 2$ and prints the largest *proper* divisor m of n. Example: For $n = 60$, your program should print $m = 30$. For $n = 61$, your program should print $m = 1$. **(10)**

```
#include <stdio.h>
main()
{
    int n, m;
    printf("Enter an integer n >= 2 :  "); scanf("%d",&n);
```

```
    printf("The desired divisor is m = %d\n",m);
}
```