M S Mandal
27/9/12

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Department of Computer Science & Engineering

Programming and Data Structures (CS11001)

Midsem (Autumn, 1st Year)

Date: Thurs, Sep 27, 2012 — *Time:* 09:00-11:00am

Students: 660 — *Marks:* 60

Answer ALL the questions.

Only the answers on the question paper itself in the spaces provided will be evaluated.

Roll no: ____________________ **Section:** _______ **Name:** ______________________________________

1. (a) Convert the following for-loop into the corresponding while-loop:

```
for (i = 2; i <= sqrt (n); i += 3)
 { x = x + y * i;
   y = y - 1;
 }
```

3

(b) Represent the integer -520 in sign-magnitude and 2's complement representations *using minimum number of bits*. Show the steps of your calculations. 6

-520 in sign-magnitude representation is ______________________________,

and in 2's complement representation is ______________________________.

(c) Represent 67.375 $\times 2^{47}$ in IEEE 754 single precision format. Show the steps of your calculation clearly:

6

(d) Let T be a function from the set of non-negative numbers to itself. Let it be defined as follows:

$T(n) = 1,$ for $n \leq 2$

$= 3 \times T(n/3) + n,$ for $n > 2$, where '/' stands for integer division.

Show that $T(n) = O(n \cdot log_3 n)$.

5

2. What does the following program compute?:

```
#include <stdio.h>
main()
 { int x, d, f;
   int n, i;
   printf ("feed the number of numbers:");
   scanf ("%d", &n);
   printf ("feed the first two numbers:");
   scanf ("%d, %d", &d, &f);
   if (d < f) { i = d; d = f; f = i;}
   for (i = 0; i < n - 2; i++) {
       printf ("feed the next number:");
       scanf ("%d", &x);
       if (x < f)
         { d = f; f = x;}
        else if (x < d) d = x;
     }
   printf ("left with:%d\n", d);
 }
```

Hint: Hand execute for some typical inputs and hence infer what it does in general. Partial marks will be given if you can choose a relevant set of inputs and show the results correctly for this chosen set. 5

3. Write a C program which reads an integer $n > 10$ and a real variable $x, 0 \leq |x| \leq 1$, and computes and prints the sum of the series up to n terms for the value of x read:
$sin^{-1}(x) = x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{x^5}{5} + \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdot \frac{x^7}{7} + \cdots + \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \times \cdots \times \frac{2i-3}{2i-2} \cdot \frac{x^{2i-1}}{2i-1} + \cdots$
The program should NOT use any function from the `math` library; in each step, the term to be added should be obtained from the previous term. 5

4. What does the following function return?

```
int d (int m, int n)
{ // m and n are non-negative integers
  int val = 0;
  while (m > 0)
   { if (m%2 == 1) val = val + n;
     m = m/2;
     n = 2*n;
   }
  return val;
}
```

Hint: hand execute for m = 26, n = 10. 5

5. The following program generates the series of numbers
1, 11, 21, 1121, 112112, 21122112, 2112221221, 112211231221,...
that is, given any number, n say, of the series, its successor n^+ is obtained by examining the digits of n from the least significant digit (lsd) position onward. The new number n^+ contains the digits extracted from n followed by the number of consecutive occurrences of the digit extracted in n. At any point, let d be the lsd of n and let it occur c_d number of times from this lsd onward; note that c_d does not reflect the number of other occurrences of d, if any, in n. First d is *accommodated as the lsd of n^+ pushing its other digits computed so far to the respective higher significant positions*; then c_d is *similarly accommodated in n^+*, as its new lsd. Hence, the digits of n occur in reverse sequence in n^+ interspersed with their respective counts of consecutive occurrences.
For example consider the fourth number 1121 = n. The next number n^+ is obtained as follows: the lsd of n is 1; so the most significant digit of n^+ is 1; since the next digit of n is 2, there is only one consecutive occurrence of 1 in n; so the next digit of n^+ is 1. Thus *eventually, after all the digits of n are accounted for* the first two digits of n^+ become 1 and 1. Similarly, the next digit of n is 2; so, the next significant digit of n^+ is 2; it occurs consecutively only once; so the digit of n^+ next to 2 is 1. Finally, the next digit of n is 1 and it occurs consecutively two times; so the next two digits of n^+ will be 1 followed by 2. Thus, n^+ becomes 112112 (which is interpreted as: "the digits of n from lsd onward are — 1 one time, 2 one time and 1 two times").
Fill in the blanks in the following program so that it generates this series:

```
#include <stdio.h>
main()
 { int till, pres = 1; // the present number in the series initialized
   int next;           // the next number in the series
   int  digit, nextDigit;
   printf("till how many terms of the series?:");
   scanf ("%d", &till);

   next = _____;  // next number -- initialized

   while (till ______ 0)
    { printf ("%d, ", pres);

      digit = ____________________;  // extract lsd of pres


      next = _________________________; // accommodate "digit"
      next = next * 10 + 1;         // accommodate its count so far

      pres /= ______;    /* place the next higher significant digit of
                            "pres" in its  lsd */

      /* now count the number of consecutive occurrences
         of the last digit introduced in "next" */

      while (pres != 0)
       { nextDigit = ____________; // get next digit of "pres"

         while (nextDigit ___ digit && pres != 0)
          { next++;              // increase the count of digit
```

```
           pres _________;  // put next digit of "pres" in its lsd

           nextDigit = __________;
         } // counting consecutive occurrences

         //control here with nextDigit != digit OR pres == 0
        if (pres != 0)
         { // next digit != digit
           digit = nextDigit;
           next = next * 10 + digit; // "digit" (= nextDigit) in "next"
           next = next * 10 + 1;     // put its count (= 1) in "next"
           pres /= 10;
         }

      } // while pres != 0

     pres = ______; // take up "next" as the present member
     next = 0;      // and the "next" (successor to "pres") initialized
     till--;
   } /* as long as till not 0 */
  printf ("\n");
}
```

10

6. Consider the following C program, and assuming that the user always enters a word with less than 20 characters, explain what this program performs.

```
 #include <stdio.h>
 void myfunc(char *s);

 int main ()
  {
    char c, s[20];
    int i = 0;
    while ((c = getchar())!='\n') s[i++] = c;
    s[i] = c;
    myfunc(s);
    return 0;
  }

void myfunc (char *s)
 {
   if (*s != '\n') myfunc (s+1);
 }
   printf("%c",*s);
 }
```

6

7. (a) Write a C function
`void multTerm (int f[], int deg, int coef, int p[])`,
which modifies `p[i+deg]` by adding to its present content the product of `f[i]` and `coef`, $0 \leq i \leq 5$. 3

(b) Write a C function
`void polymul(int f[], int g[], int p[])`,
where the arrays `f` and `g` contain the coefficients of the polynomials $f(x)$ and $g(x)$, both of the form $a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, where each a_i is an integer $(|a_i| \geq 0)$. The function computes the coefficients of the product polynomial (of $f(x)$ and $g(x)$) in the array `p`. This function should use the function `multTerm`. 6

No need to write the `main` *function.*

Rough Work