

Indian Institute of Technology Kharagpur
Department of Computer Science & Engineering

CS10001 Programming and Data Structures

Mid semester Examination
Autumn 2008

Duration: 2 hrs

Total Marks: 30

This question paper contains SIX pages.

Name:

Roll No:

Section:

All the answers must be provided in the question paper in the designated boxes. Do your rough work on reverse side of the question sheet. Do not leave the examination hall without submitting the question paper. Write your roll no on top of every page of the question paper.

Question 1 (13)	2 (5)	3 (7)	4 (5)	Total (30)

Roll No:

1.1. What is printed by the following program?

[3]

```
int f(int x, int y) {
    int sum = 0 ;
    y-- ;
    if ( x == 0 ) return 0 ;
    else{
        printf( "%d ",x ) ;
        sum = y + f ( x - 1, y );
        printf ( "%d ", sum ) ;
    }
    return sum;
}

main( ){
    printf ( "\n %d \n", f ( 3, 8 ) ) ;
}
```

3 2 1 5 11 18 18

1.2. What output do you expect from the following program?

[3]

```
main( ){
int numbers[10]={1, 0, 0, 0, 0};
int i, j;

for (j = 0; j < 5 ; ++j)
    for (i = 0; i < j; ++i)
        numbers[j] += numbers[i];

for (j = 0; j < 5; ++j)
    printf("%d ", numbers[j]);
}
```

1 1 2 4 8

Roll No:

1.3. What will be printed by the following program?

[3]

```
void foo (char s [ ]) {
    while ( *s != '-' ) s ++ ;
    while ( *s ) {
        *s = *( s + 1 ) ;
        s++ ;
    }
}
main ( ) {
    char s[20] = "I-I-T-Kgp";

    foo(s);
    printf("%s",s);
}
```

I-T-Kgp

1.4 What will be printed on executing the following program?

[3]

```
int x=100;
void foo1 (int y) {
    y++;
    printf("foo1: x=%d\n", x) ;
    x++;
}
void foo2 (int z) {
    static int x=0;
    x++;
    printf ("foo2: x=%d\n", x) ;
}
void foo3 ( ) {
    printf("foo3: x=%d\n", x) ;
}
int main ( ) {
    int x;
    x = 20;
    foo1 (x) ;
    foo2(x) ;
    foo2 (x) ;
    foo3 (x) ;
    printf ("Main: x=%d\n", x) ;
}
```

foo1: x=100
foo2: x=1
foo2: x=2
foo3: x=101
Main: x=20

Roll No:

2. Write a program which reads an integer n ($n \leq 20$), and then reads two arrays a and b, each containing n distinct integer elements. Considering each array as a set of integers, the program should print the integers that are common to the two sets. [5]

Roll No:

3. Write a program which will find out indices (row and column) of all the “saddle points” in a two dimensional integer array of size $n \times n$ ($n < 100$). An array element is a saddle point if it is the “maximum” in its column and the “minimum” in its row. Assume all array elements as having distinct values. For example (1,1) is the saddle point of the 2-D array – [5+2]

```

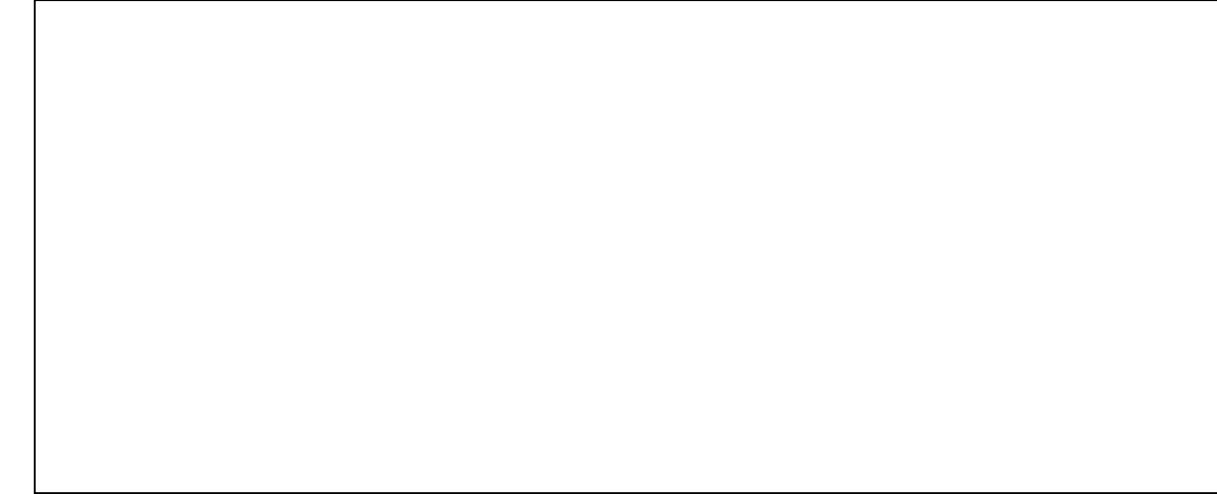
1 2 3
7 5 6
8 4 9

```

/* The function is_saddlepoint takes as input a matrix M, its size n, and the row and column indices i and j of an element. It returns 1 if M[i][j] is a saddle point, and 0 otherwise. */

```
int is_saddlepoint (
```

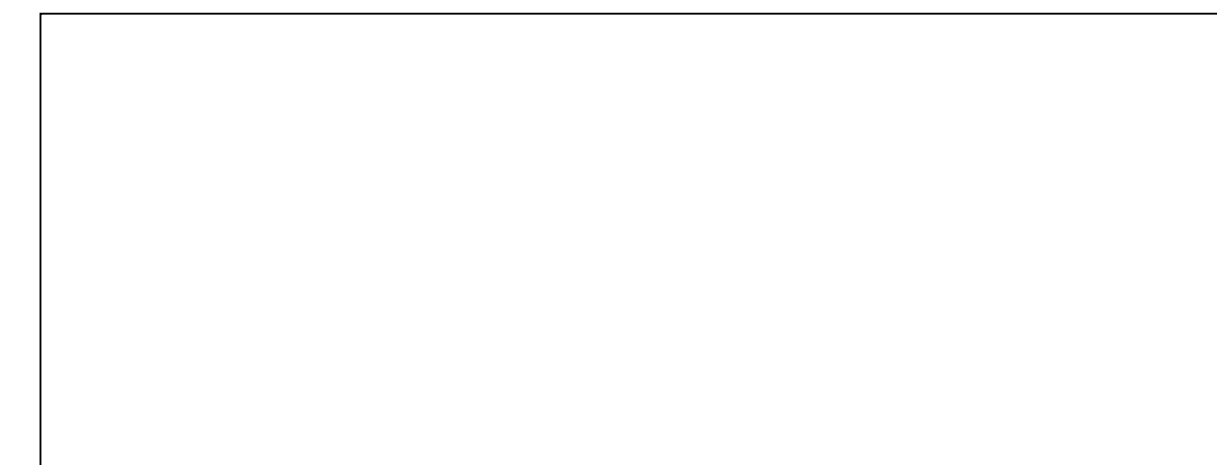
```
{
```



```
}
```

/* main should identify the saddle points by making calls to the function is_saddlepoint(..) */

```
int main ( ) {
    int row, col, n, M [ 100 ] [ 100 ] ;
    scanf ( " %d ", &n ) ;
    for ( row = 0 ; row < n ; row ++ )
        for ( col = 0 ; col < n ; col ++ )
            scanf ("%d", & M [ row ] [ col ] );
```



```
}
```

Roll No:

4. For an integer k we have

[5]

$$a^{2k} = (a^k)^2, \text{ and}$$

$$a^{2k+1} = (a^k)^2 * a.$$

Use this observation to write a **recursive** function that, given a positive real number a and a non-negative integer n , computes a^n . The function should be efficient in terms of the total number of multiplications required to find the answer.

END