

**Lab. Exercises #8 Solution (Lab Date: Week 10 of Classes)**

Objectives are to:

1. Practise on the use of nested looping with functions and arrays in problem solving as taught in chapters 7 and 8. Also, practise on the use of flowcharts.
2. In particular, to learn how to specify array parameters in function prototypes, function calls and function definitions. Also, to learn how to specify array parameters as call-by-value and as call-by-reference parameters.
3. Continue to practise for quiz #2 if not yet written.

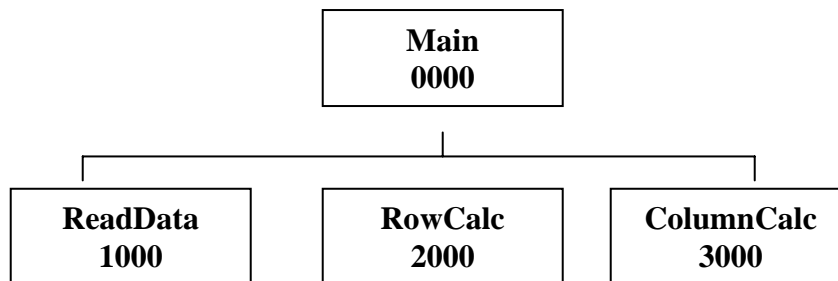
Que. 1. Assume you have the following table of scores,

90	80	75	98	100	91
50	65	70	80	90	67
69	77	68	71	50	55

write a program using top-down design to read this table, print the average and maximum value of each row and column.

**Solution to Que 1.**

Solve using the following structure chart.



- Use the ReadData function to read the two dimensional array table.
- Use the RowCalc function to navigate the rows and obtain the minimum element, total and average of each row.
- Use the ColCalc function to navigate the columns and obtain the minimum element, total and average of each column.

Now, complete the solution to the problem above by including the function prototypes for the other two functions RowCalc and ColumnCalc. Then, make proper function calls to these functions in the main driver and provide the function definitions for the functions following the ReadData function definition.

Script started on Sat Sep 04 20:47:05 2010

sol:~/bk2010/programs>cat lab8slnq1.c

```
#include <stdio.h>
```

```
#define ROW 3
```

```
#define COL 6
```

```
void ReadData (int [][][COL]);
```

```
void RowCalc (int [][][COL]);
```

```
void ColumnCalc (int [][][COL]);
```

```
/* Que. 1. Assume you have the following table of scores,
```

```
90      80      75      98      100     91
```

```
50      65      70      80      90      67
```

```
69      77      68      71      50      55
```

write a program using top-down design to read this table, print the average and maximum value of each row and column.

```
*/
```

```
int main(void){
```

```
    int score[ROW][COL];
```

```
    ReadData (score);
```

```
    RowCalc (score);
```

```
    ColumnCalc (score);
```

```
    return 0;
```

```
}
```

```
void ReadData (int score[][COL]){
```

```
    int r, c;
```

```
    printf("Type scores for each row on a new line:\n");
```

```
    for (r=0; r<ROW; r++)
```

```
        for (c=0; c<COL; c++)
```

```
            scanf("%d", &score[r][c]);
```

```
}
```

```
void RowCalc (int score[][COL]){
```

```
    int r, c, sum, max;
```

```
    float avg;
```

```
    printf("          ");
```

```
    for (c=0; c<COL; c++)
```

```
        printf("Col%-3d ", c);
```

```
    printf("RowAvg RowMax\n");
```

```
    for (r=0; r<ROW; r++) {
```

```
        printf ("Row%d   ", r + 1);
```

```
        sum = 0;
```

```
        max = score[r][0];
```

```
        for (c=0; c<COL; c++) {
```

```
            printf ("%6d ", score[r][c]);
```

```
            sum += score[r][c];
```

```
            if (score[r][c] > max)
```

```
                max = score[r][c];
```

```

    }
    avg = (float)sum/COL;
    printf ("%6.2f %6d\n", avg, max);
}
}

void ColumnCalc (int score[][COL]){
    int r, c, col_sum[COL], col_max[COL];
    float col_avg[COL];
    for (c=0; c<COL; c++) {
        col_sum[c] = 0;
        col_max[c] = score[0][c];
        for (r=0; r<ROW; r++) {
            col_sum[c] = col_sum[c] + score[r][c];
            if (score[r][c] > col_max[c])
                col_max[c] = score[r][c];
        }
        col_avg[c] = (float)col_sum[c]/ROW;
    }
    printf ("\nColAvg");
    for (c=0; c<COL; c=c+1)
        printf (" %-6.2f", col_avg[c]);
    printf ("\nColMax ");
    for (c=0; c<COL; c=c+1)
        printf ("%6d ", col_max[c]);
    printf ("\n");
}

```

sol:~/bk2010/programs>cc lab8slnq1.c

sol:~/bk2010/programs>a.out

Type scores for each row on a new line:

90 80 75 98 100 91

50 65 70 80 90 67

69 77 68 71 50 55

	Col0	Col1	Col2	Col3	Col4	Col5	RowAvg	RowMax
Row1	90	80	75	98	100	91	89.00	100
Row2	50	65	70	80	90	67	70.33	90
Row3	69	77	68	71	50	55	65.00	77

ColAvg 69.67 74.00 71.00 83.00 80.00 71.00

ColMax 90 80 75 98 100 91

sol:~/bk2010/programs>exit

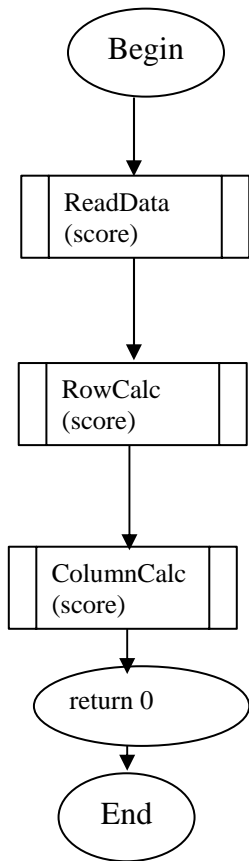
exit

script done on Sat Sep 04 20:47:37 2010

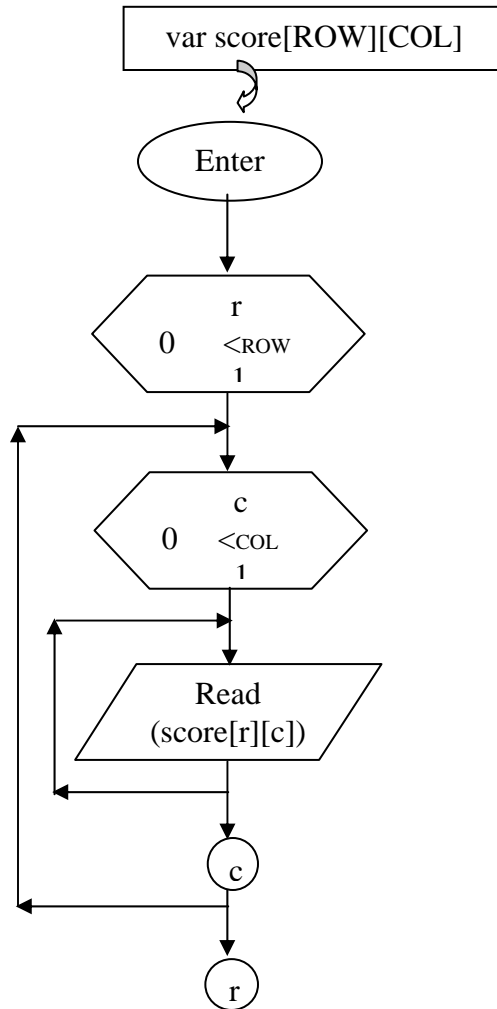
The flowcharts for the above solution is also provided below.

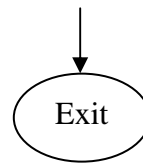
The flowchart:

0000  
Control Module

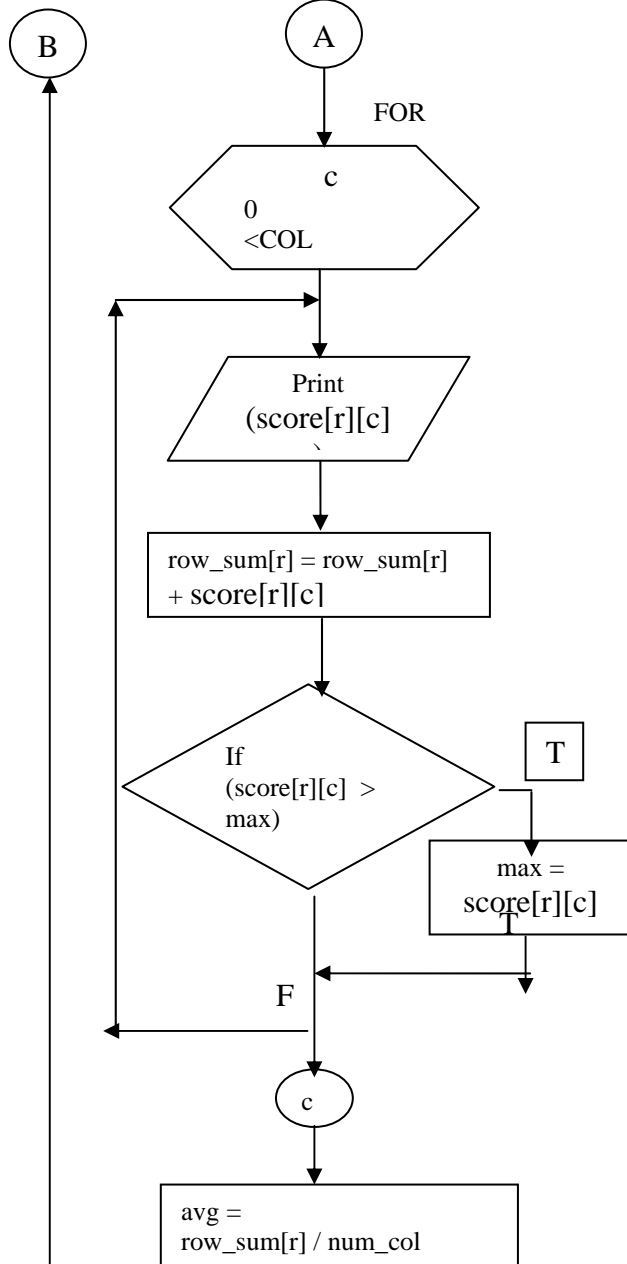
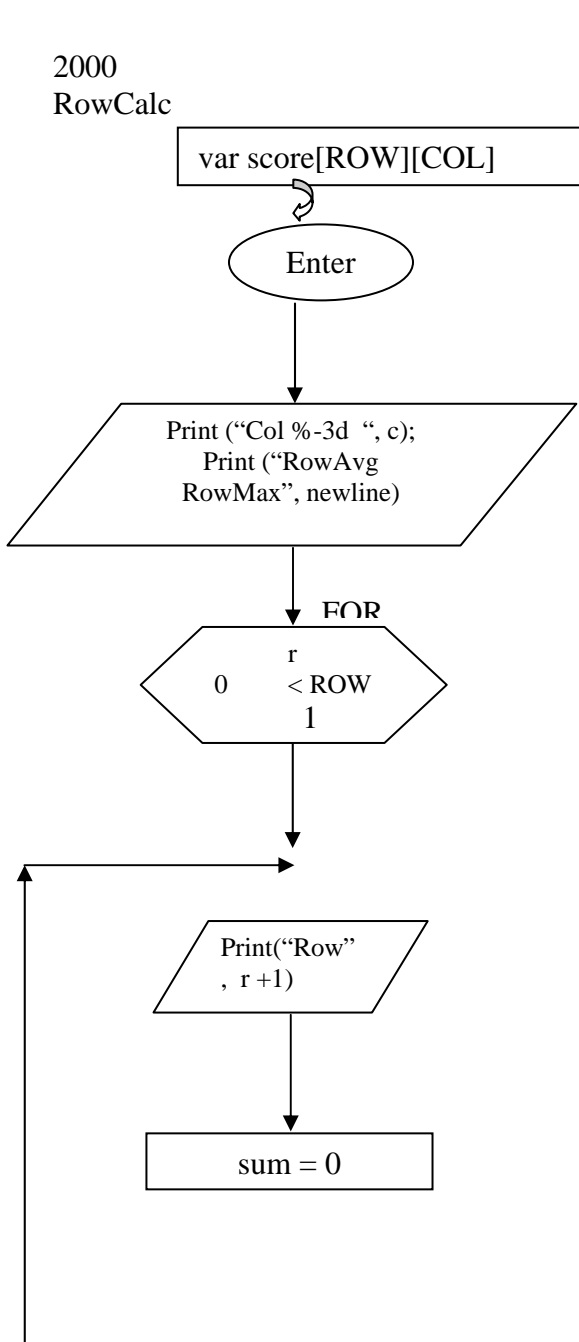


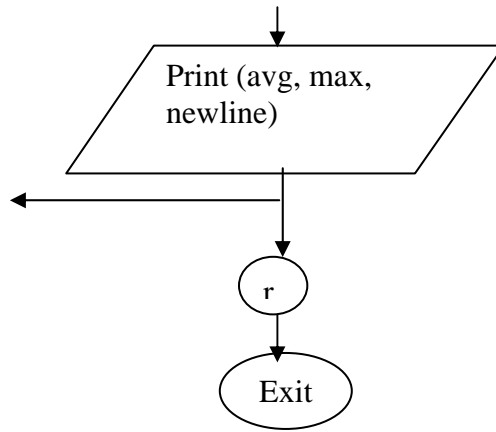
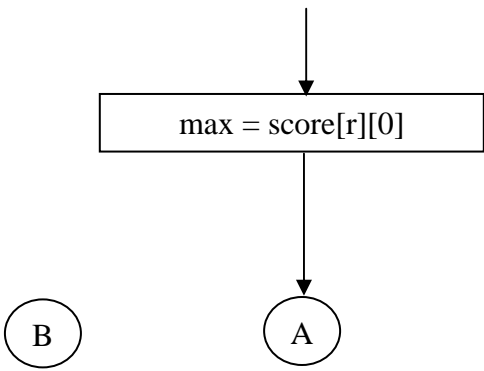
1000  
ReadData



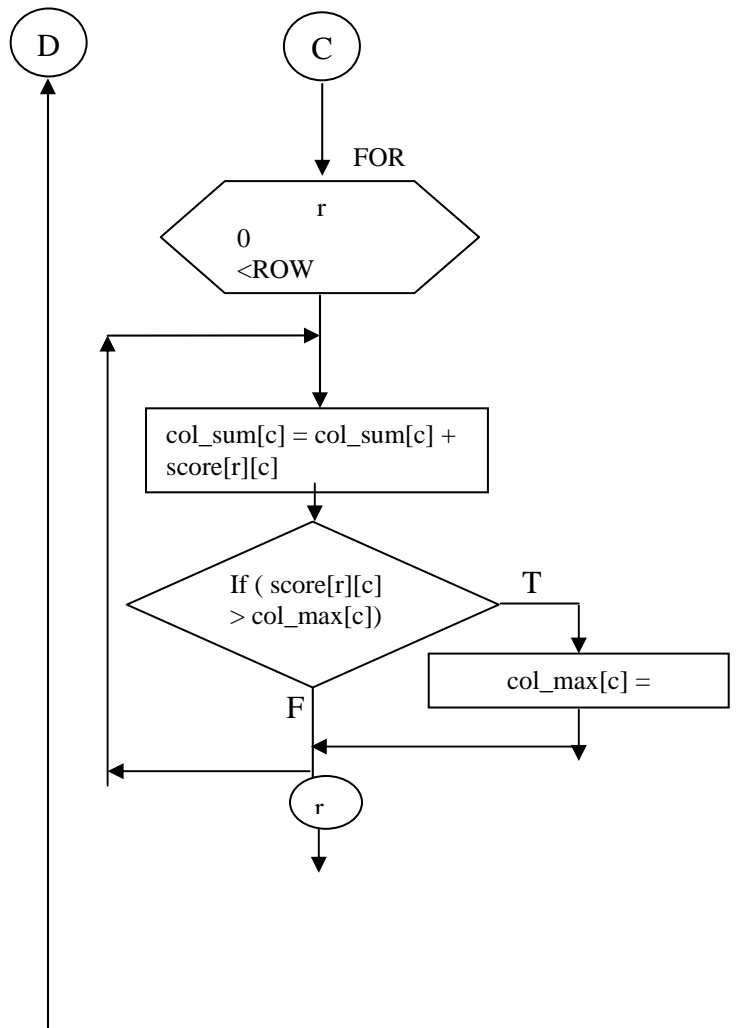
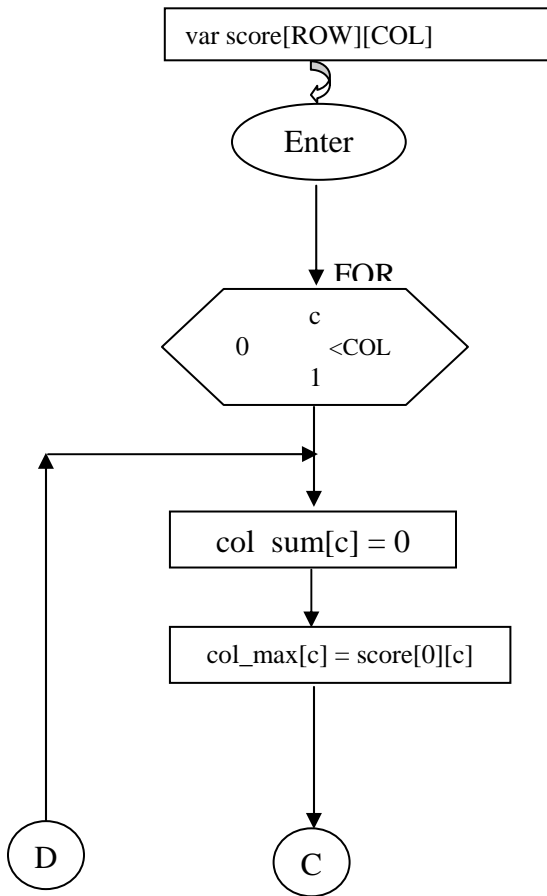


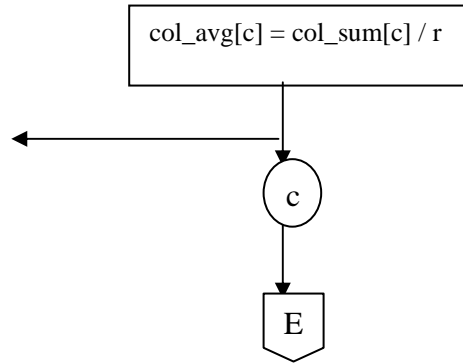
2000  
RowCalc



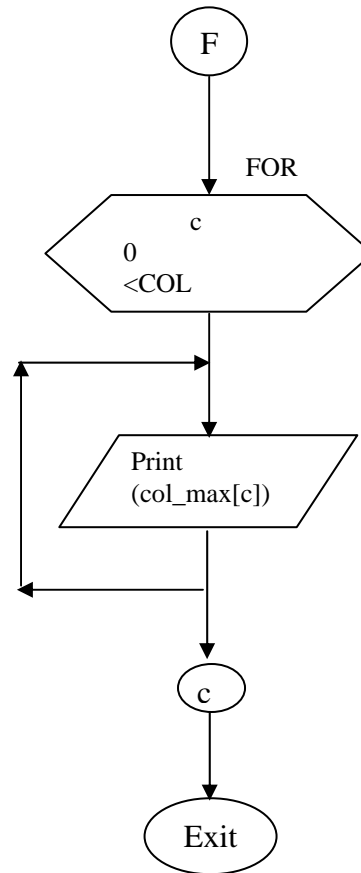
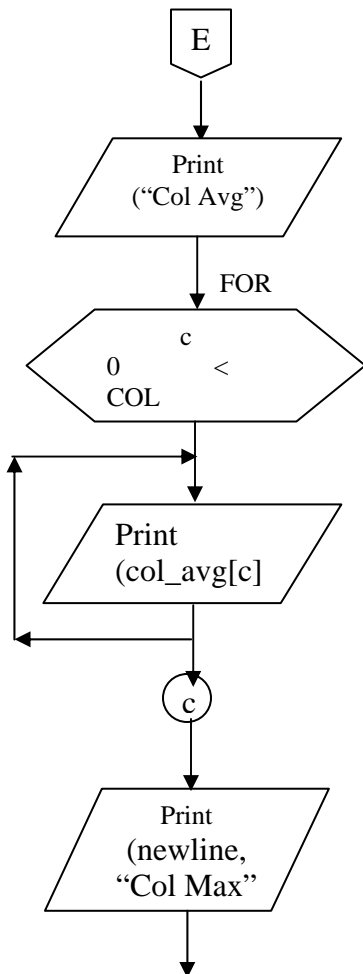


3000  
ColumnCalc





3000  
ColumnCalc Cont'



F

Que. 2. (Optional) Define a problem of your own that involves loops and solve the same way as in question 1.