GE3 COMPUTER SCIENCE

C AND C ++ LECTURE SERIES *FOR*

B.SC 3RD SEMESTER *BY*

SUBHADIP MUKHERJEE

DEPARTMENT OF COMPUTER SCIENCE

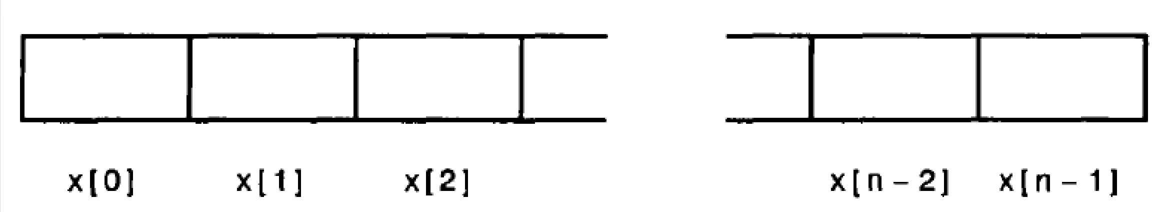
KHARAGPUR COLLEGE

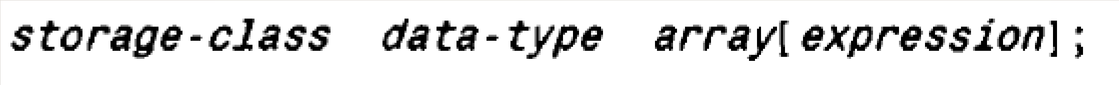
LECTURE 10

ARRAY

DEFINITION

- Arrays are defined in much the same manner **as** ordinary variables, except that each array name must be accompanied by a size specification (i.e., the number of elements).

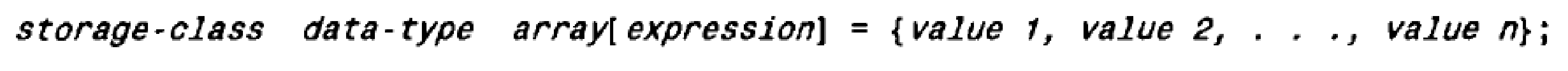




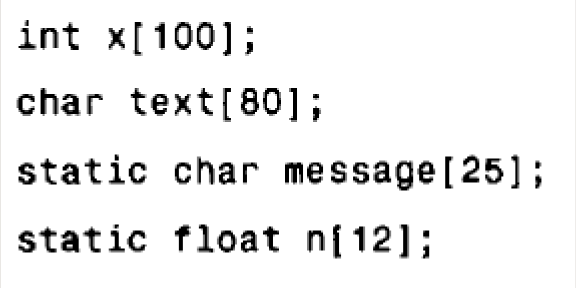
```
storage-class   data-type   array[ expression];
```

ARRAY

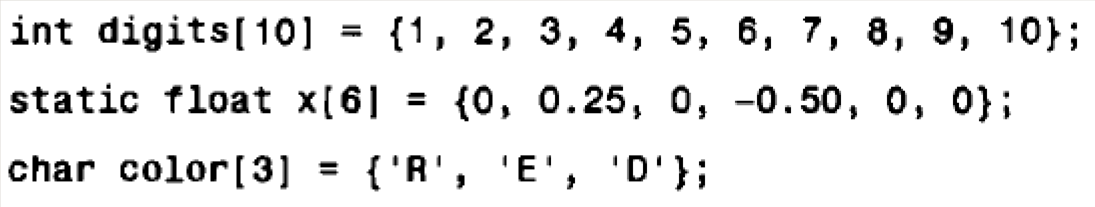
EXAMPLE



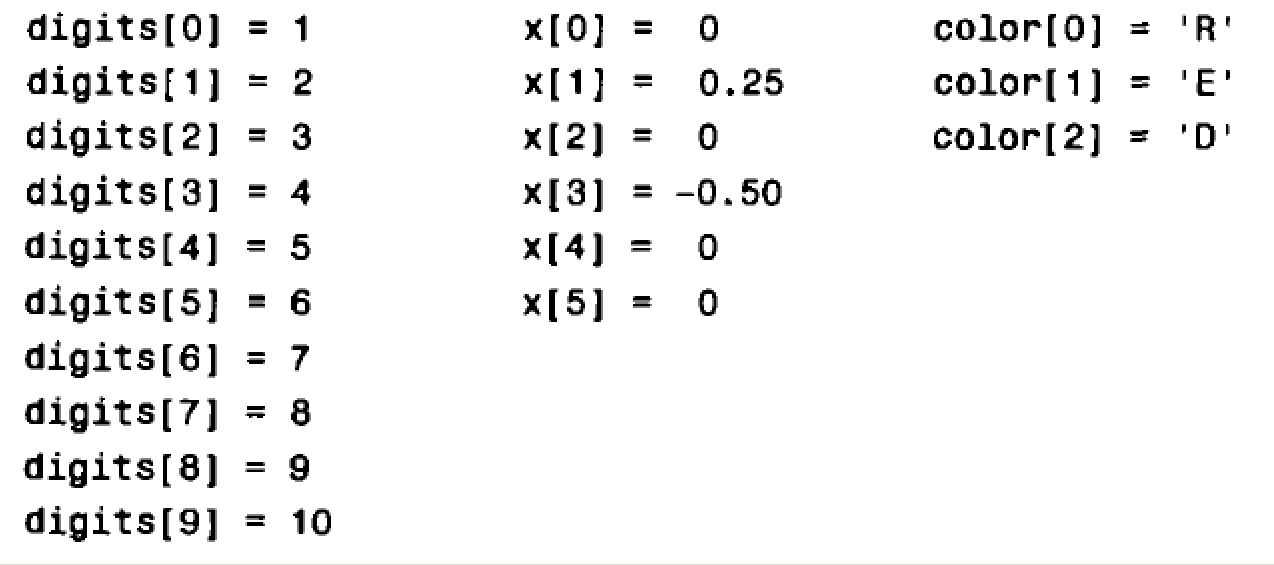
```
storage-class  data-type  array[expression] = {value 1, value 2, . . ., value n};
```



```
int x[100];
char text[80];
static char message[25];
static float n[12];
```



```
int digits[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
static float x[6] = {0, 0.25, 0, -0.50, 0, 0};
char color[3] = {'R', 'E', 'D'};
```

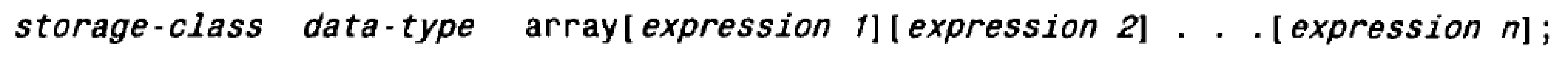


```
digits[0] = 1          x[0] =  0          color[0] = 'R'
digits[1] = 2          x[1] =  0.25       color[1] = 'E'
digits[2] = 3          x[2] =  0          color[2] = 'D'
digits[3] = 4          x[3] = -0.50
digits[4] = 5          x[4] =  0
digits[5] = 6          x[5] =  0
digits[6] = 7
digits[7] = 8
digits[8] = 9
digits[9] = 10
```

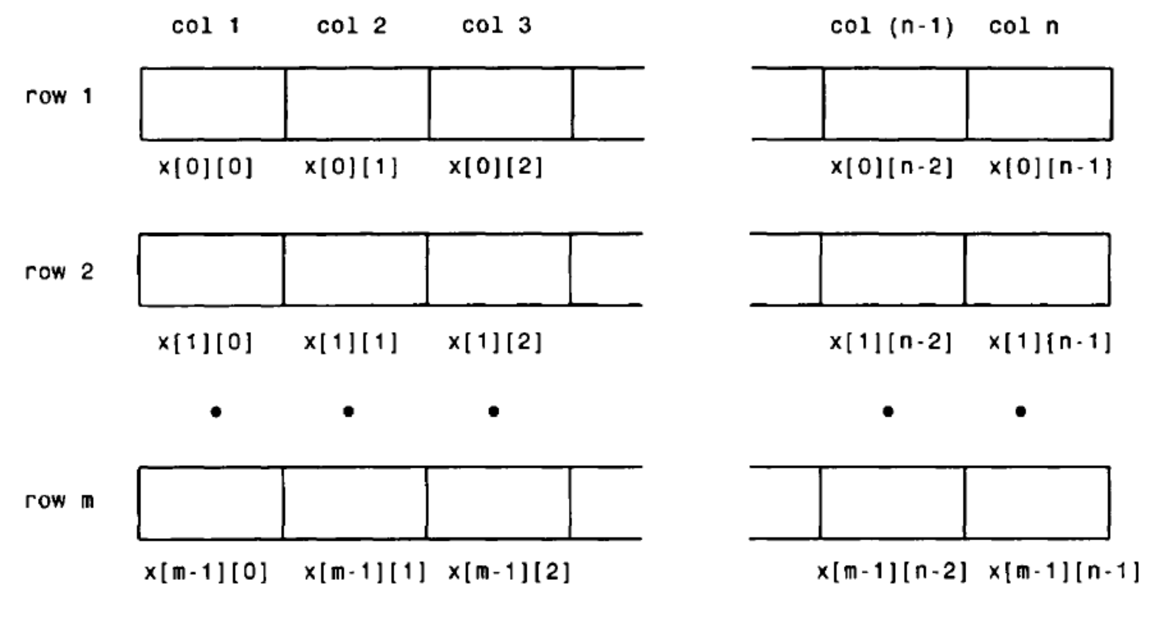
ARRAY

MULTIDIMENSIONAL ARRAYS

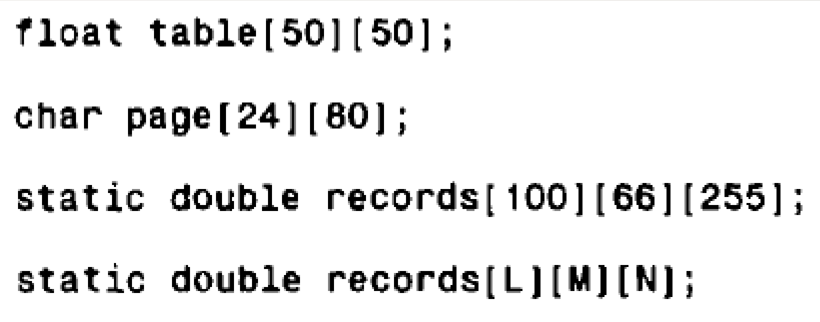
- Multidimensional arrays are defined in much the same manner as one-dimensional arrays, except that a separate pair of square brackets is required for each subscript



```
storage-class  data-type  array[expression 1][expression 2] . . .[expression n];
```



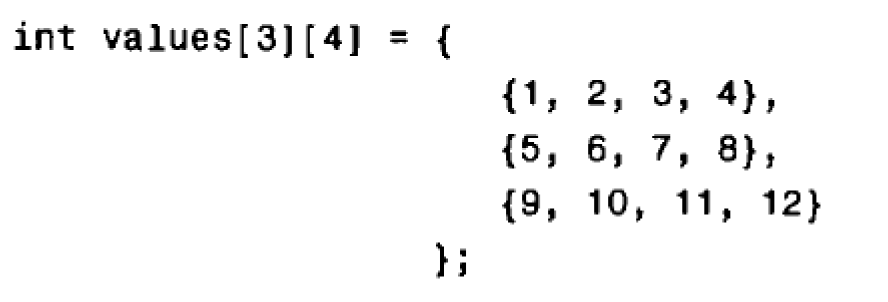
	col 1	col 2	col 3		col (n-1)	col n
row 1	x[0][0]	x[0][1]	x[0][2]		x[0][n-2]	x[0][n-1]
row 2	x[1][0]	x[1][1]	x[1][2]		x[1][n-2]	x[1][n-1]
	•	•	•		•	•
row m	x[m-1][0]	x[m-1][1]	x[m-1][2]		x[m-1][n-2]	x[m-1][n-1]



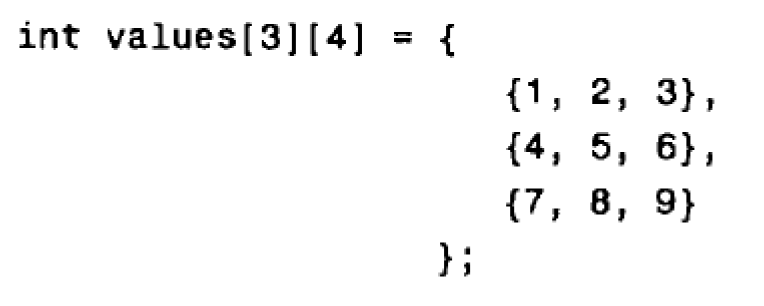
```
float table[50][50];

char page[24][80];

static double records[100][66][255];

static double records[L][M][N];
```

ARRAY

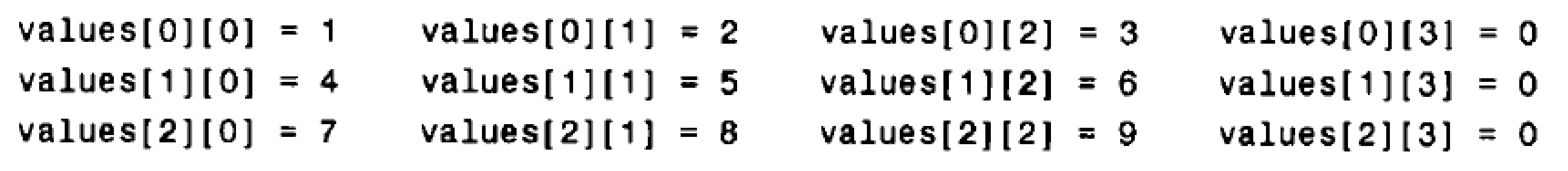
MULTIDIMENSIONAL ARRAYS



```
int values[3][4] = {
                        {1, 2, 3, 4},
                        {5, 6, 7, 8},
                        {9, 10, 11, 12}
                     };
```



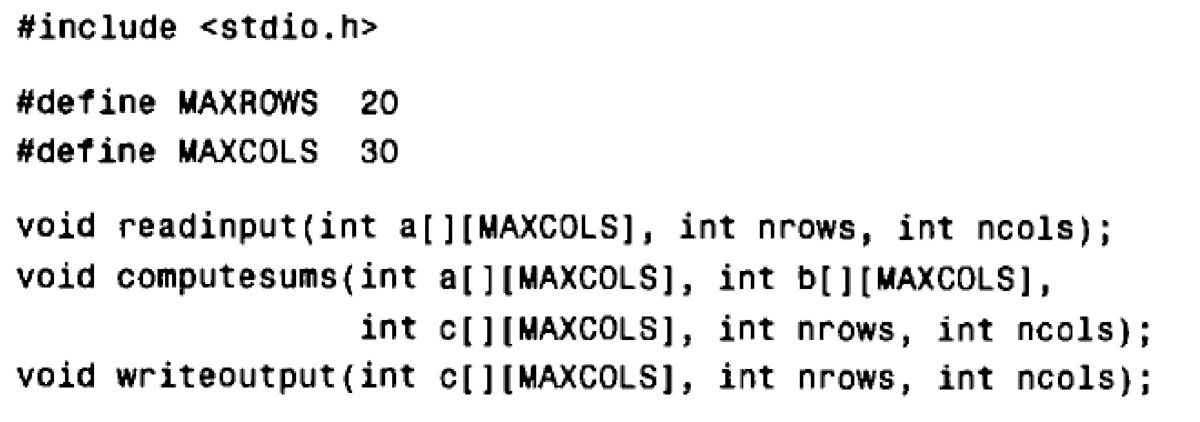
```
int values[3][4] = {
                        {1, 2, 3},
                        {4, 5, 6},
                        {7, 8, 9}
                     };
```



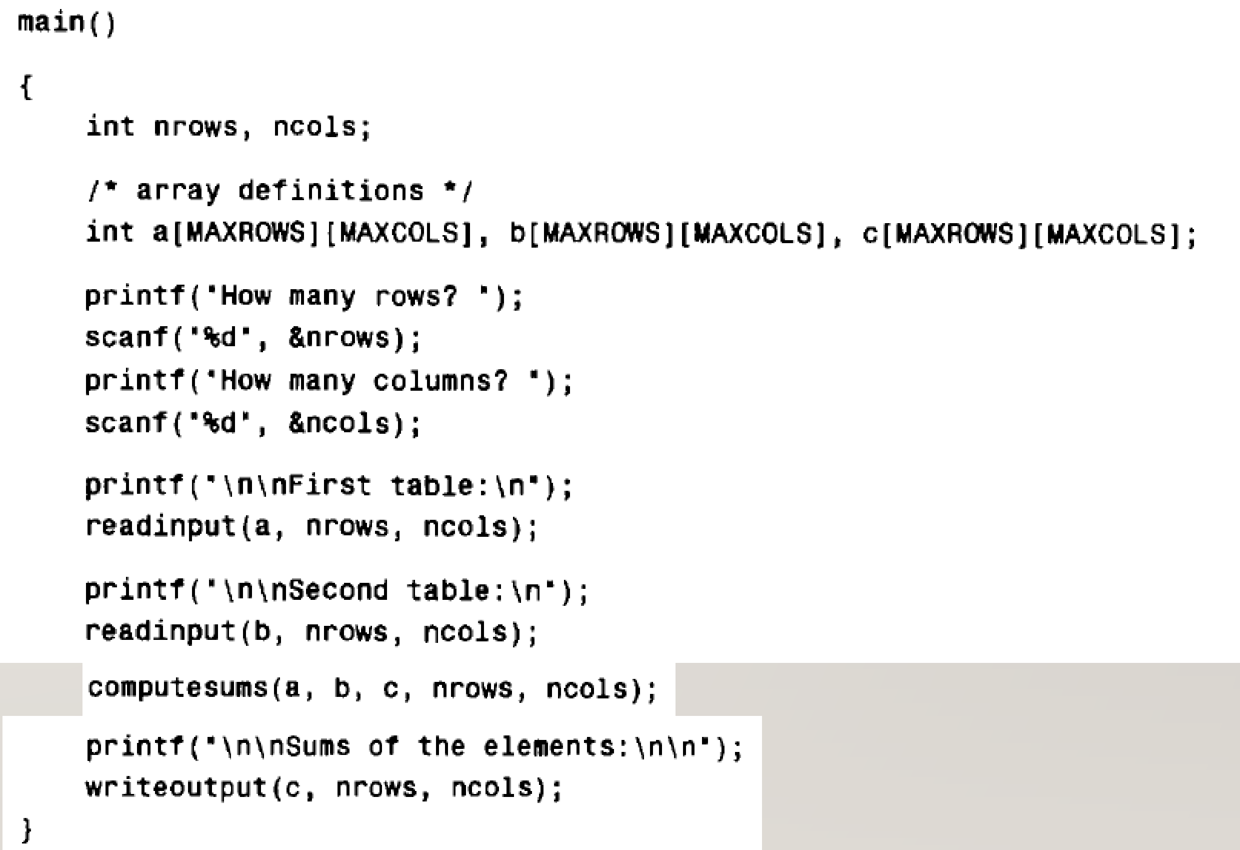
```
values[0][0] = 1    values[0][1] = 2    values[0][2] = 3    values[0][3] = 0
values[1][0] = 4    values[1][1] = 5    values[1][2] = 6    values[1][3] = 0
values[2][0] = 7    values[2][1] = 8    values[2][2] = 9    values[2][3] = 0
```

ARRAY

Add Two Matrices (Add two tables)



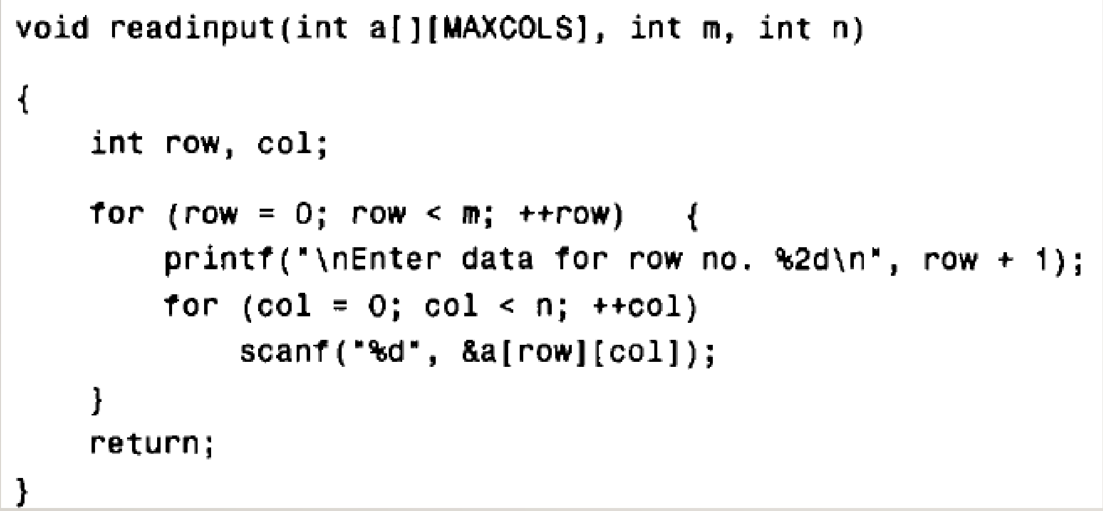
```
#include <stdio.h>

#define MAXROWS  20
#define MAXCOLS  30

void readinput(int a[][MAXCOLS], int nrows, int ncols);
void computesums(int a[][MAXCOLS], int b[][MAXCOLS],
                 int c[][MAXCOLS], int nrows, int ncols);
void writeoutput(int c[][MAXCOLS], int nrows, int ncols);
```



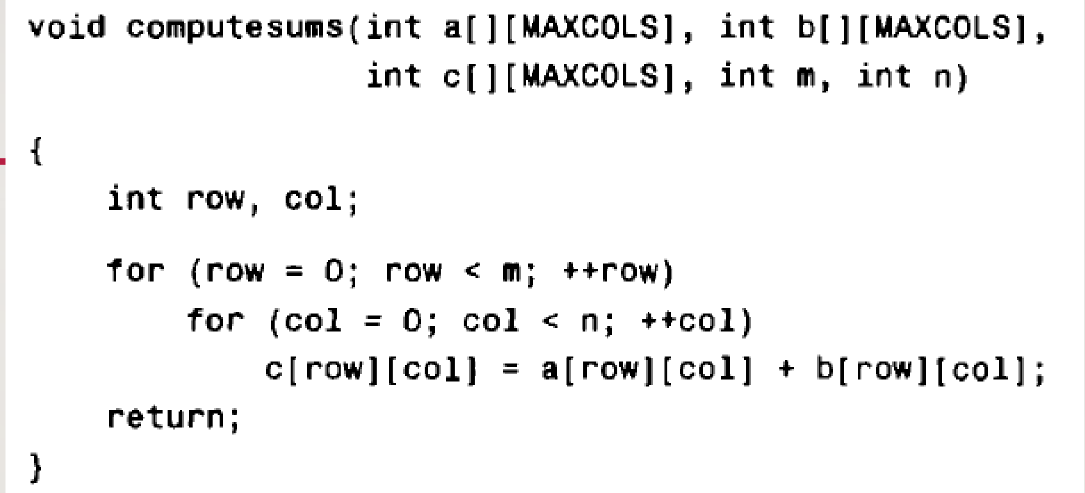
```
main()

{
    int nrows, ncols;

    /* array definitions */
    int a[MAXROWS][MAXCOLS], b[MAXROWS][MAXCOLS], c[MAXROWS][MAXCOLS];

    printf("How many rows? ");
    scanf("%d", &nrows);
    printf("How many columns? ");
    scanf("%d", &ncols);

    printf("\n\nFirst table:\n");
    readinput(a, nrows, ncols);

    printf("\n\nSecond table:\n");
    readinput(b, nrows, ncols);

    computesums(a, b, c, nrows, ncols);

    printf("\n\nSums of the elements:\n\n");
    writeoutput(c, nrows, ncols);
}
```

ARRAY

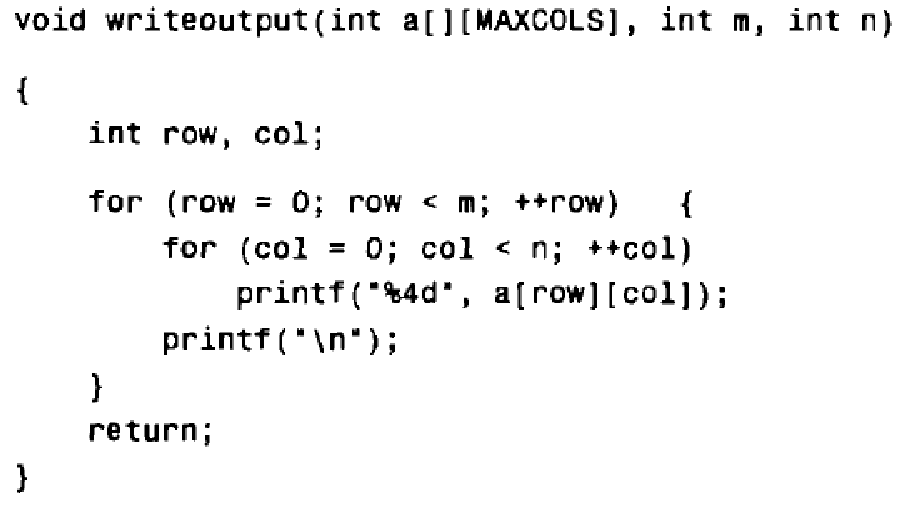
Add Two Matrices (Add two tables)



```c
void readinput(int a[][MAXCOLS], int m, int n)

{
    int row, col;

    for (row = 0; row < m; ++row)   {
        printf("\nEnter data for row no. %2d\n", row + 1);
        for (col = 0; col < n; ++col)
            scanf("%d", &a[row][col]);
    }
    return;
}
```



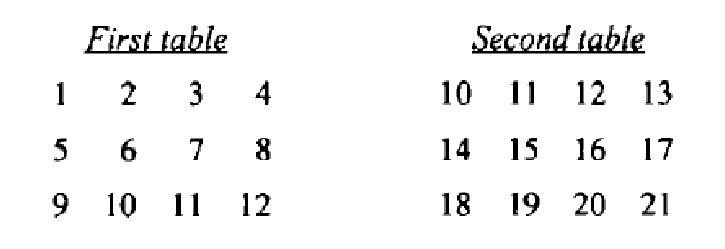
```c
void computesums(int a[][MAXCOLS], int b[][MAXCOLS],
                 int c[][MAXCOLS], int m, int n)

{
    int row, col;

    for (row = 0; row < m; ++row)
        for (col = 0; col < n; ++col)
            c[row][col] = a[row][col] + b[row][col];
    return;
}
```



```c
void writeoutput(int a[][MAXCOLS], int m, int n)

{
    int row, col;

    for (row = 0; row < m; ++row)   {
        for (col = 0; col < n; ++col)
            printf("%4d", a[row][col]);
        printf("\n");
    }
    return;
}
```

ARRAY

DEFINITION

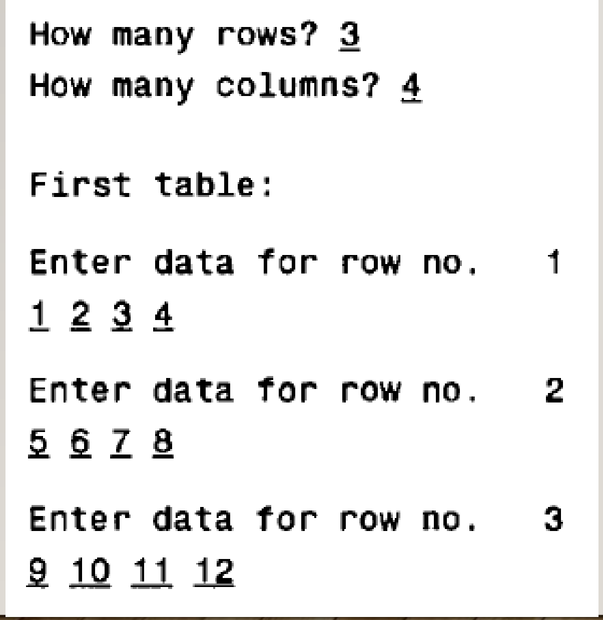


First table

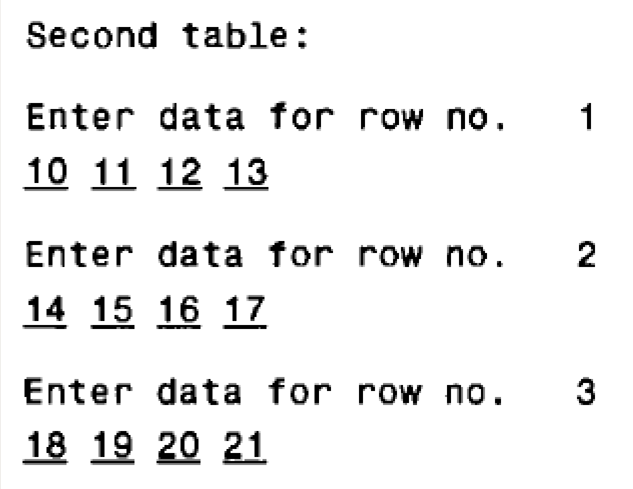
1	2	3	4
5	6	7	8
9	10	11	12

Second table

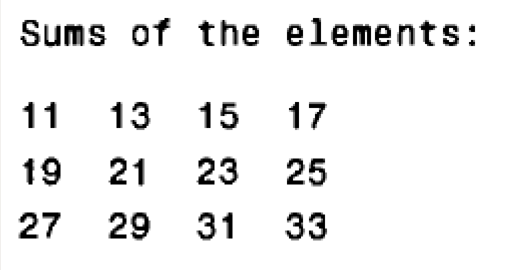
10	11	12	13
14	15	16	17
18	19	20	21



```
How many rows? 3
How many columns? 4

First table:

Enter data for row no.   1
1 2 3 4

Enter data for row no.   2
5 6 7 8

Enter data for row no.   3
9 10 11 12
```



```
Second table:

Enter data for row no.   1
10 11 12 13

Enter data for row no.   2
14 15 16 17

Enter data for row no.   3
18 19 20 21
```



```
Sums of the elements:

11   13   15   17
19   21   23   25
27   29   31   33
```

COMPILE AND RUN A C CODE

Thank You

End of Lecture 10

Subhadip Mukherjee

Department of Computer Science

Kharagpur College

Kharagpur, India