```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void access2D(int mat[3][5], int i, int j){
    printf("Number of rows and cols both known\n");
    printf("Element at index (%d,%d) is %d\n",i,j,mat[i][j]);
}

void access2DRowUnknown(int mat[][5], int i, int j){
    printf("Number of rows unknown\n");
    printf("Element at index (%d,%d) is %d\n",i,j,mat[i][j]);
}

// The following function will cause a compilation error
// Must provide number of columns otherwise Mr C does not know
// how many elements to skip from first row when going to second row
// void access2DRowColUnknown(int mat[][], int i, int j){
//     printf("Element at index (%d,%d) is %d\n",i,j,mat[i][j]);
// }

// One way to overcome this problem is to index the 2D array as a
// 1D array as it is stored in memory and find out the location of
// the [i][j] index ourselves
void access2DRowColUnknown(int *mat, int i, int j, int numCols){
    printf("Number of rows and columns both unknown\n");
    printf("Element at index (%d,%d) is %d\n",i,j,*(mat + i * numCols + j));
}

// The other way is to use arrays of arrays since every row in an
// array of arrays has a separate pointer to the first element of
// the row. So we dont need to know how many entries are in each row
void accessArrArrRowColUnknown(int **mat, int i, int j){
    printf("Number of rows and columns both unknown\n");
    printf("Element at index (%d,%d) is %d\n",i,j,mat[i][j]);
}

int main(){
    int a[3][5] = {{1,2,3,4,5},{10,20,30,40,50},{100,200,300,400,500}};
    access2D(a, 1, 0);
    access2DRowUnknown(a, 1, 2);
    int i, j, numCols = 5;
    // The following will generate a warning about
    // incompatible pointer types
    // access2DRowColUnknown(a, 2, 0, numCols);

    // To be safe, do the following
    int *ptr = &a[0][0];
    access2DRowColUnknown(ptr, 2, 0, numCols);

    int **mat = (int**)malloc(3 * sizeof(int*));
    for(i = 0; i < 3; i++){
        mat[i] = (int*)malloc(numCols * sizeof(int));
        for(j = 0; j < numCols; j++){
            mat[i][j] = (j + 1) * (int)pow(10,i);
        }
    }
    accessArrArrRowColUnknown(mat, 2, 2);
}
```