COMP20005 – Class Exercise, May 2015

Consider the following declarations:

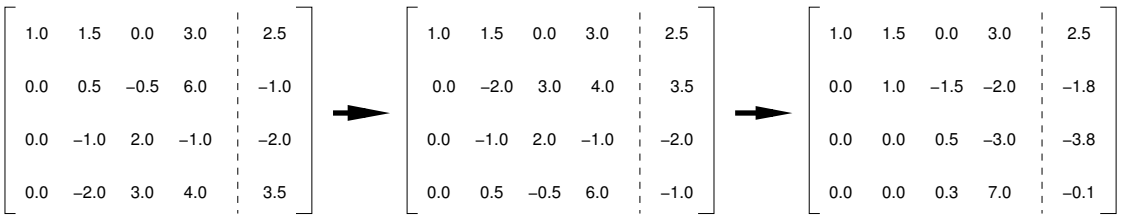
```
#define MAX 100
double A[MAX][MAX+1];
```

The two-dimensional array `A` is to be used in the process of Gaussian elimination, and has one more column than row.

The i th pivoting step of the forwards elimination process consists of a sequence of steps:

- Identify the value r in the range $i \ldots n-1$ with the largest absolute value for $A[r][i]$
- Swap all pairs of values in the i th and r th rows
- After swapping, take $A[i][i]$ to be the *pivot* value; if it is zero or close to zero, no solution is possible
- Divide each element in the i th row by the pivot value, to normalize that row and set $A[i][i]$ to one.
- For each of the rows $i+1 \leq r \leq n-1$ that are below the i th row, subtract a suitable multiple of the i th row from the r th row, so as to set $A[r][i]$ to be zero.

For example, after one pivoting step (done with $i = 0$) in a set of $n = 4$ equations, the augmented matrix representing some initial set of equations might be as shown on the left-hand side of the arrangement below.



The second pivoting step with $i = 1$ then: selects row $j = 3$ as having the largest value in the $i = 1$ th column; swaps row 1 and row 3 to get the middle arrangement; divides all of row 1 by -2.0 to normalize the element $A[1][1]$ to 1.0; and then eliminates the coefficients in $A[2][1]$ and $A[3][1]$, to yield the third arrangement.

Note that in this example all computations are shown rounded to one decimal place. In a program, `double` variables would get used.

(a) Write a function

```
void swap_matrix_rows(double A[MAX][MAX+1],
    int n, int i, int r)
```

that exchanges each value in the `i` th row of the augmented matrix `A` with the corresponding value in the `r` th row of A. The first `n` $\times$ (`n` $+ 1$) values of A have valid values.

(b) The function on the next two pages carries out the i th pivoting step, adjusting the $\mathtt{n} \times (\mathtt{n}+1)$ augmented matrix stored in the array `A` that is passed as an argument. It makes use of the function specified in part (a) of this question.

Complete the function using the boxes provided. Note that the number of boxes corresponds closely to the number of statements that need to be written; and don't forget any braces "{" and "}"that may be required. If the absolute value of the pivot element is less than 10^{-5}, the function is to print an error message and then terminate program execution.

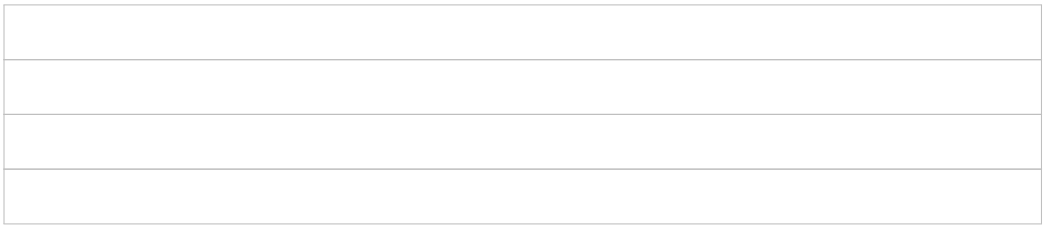
```
#include <math.h>
#define EPS 1e-5

void
pivot_matrix(double A[MAX][MAX+1], int n, int i) {
    int j, r, c;
    double pivotval;
    /* identify the pivot row */
    r = i;
```

```
    for (j=i+1; j<n; j++) {
        if (fabs(A[j][i]) > fabs(pivotval)) {
            pivotval = A[j][i];
```

```
            r = j;
        }
    }
    /* swap the i'th and r'th rows over */
```

```
    /* check that pivot value is safe, error and exit if not  */
```



```
    /* normalize the i'th row */
    for (c=i; c<=n; c++) {
        A[i][c] = A[i][c]/pivotval;
    }

    /* and now adjust the other rows below the i'th one */
```

```
    return;
}
```

(c) Write a calling function

```
void make_upper_triangular(double A[MAX][MAX+1], int n)
```

that converts the $\mathtt{n} \times (\mathtt{n}+1)$ augmented matrix stored in the array `A` in to upper-triangular form (if that conversion is possible), with 1's down the lead diagonal.

(d) Describe in words the remaining steps required to solve the system of equations, and convert A to a matrix in which the final column is the set of required values.