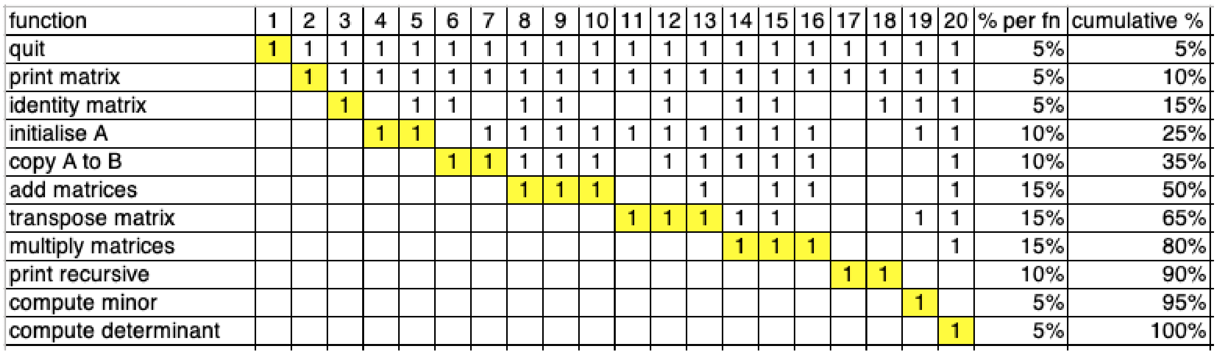
Computation I 5EIA0
Homework 3: Matlab (v2.4, September 18, 2022)
Deadline Tuesday 27 September 13:30

In this exam you will develop a competitor to the Matlab program! The user will be able to create matrices, add, transpose, and multiply them, and compute the determinant.



function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	% per fn	cumulative %
quit	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5%	5%
print matrix		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5%	10%
identity matrix			1		1	1		1	1			1		1	1			1	1	1	5%	15%
initialise A				1	1		1	1	1	1	1	1	1	1	1	1			1	1	10%	25%
copy A to B						1	1	1	1	1		1	1	1	1	1				1	10%	35%
add matrices								1	1	1			1		1	1				1	15%	50%
transpose matrix											1	1	1	1	1				1	1	15%	65%
multiply matrices														1	1	1				1	15%	80%
print recursive																	1	1			10%	90%
compute minor																			1		5%	95%
compute determinant																				1	5%	100%

Figure 1: Test cases and points per task.

Task 1. Write a C program that asks the user to select the command that needs to be performed. The commands that need to be supported are listed in the following table:

command	operation
q	quit program
I	create matrix A equal to the identity matrix
a	print matrix A
b	print matrix B
c	print matrix C
A	create matrix A
B	copy matrix A to matrix B
+	add matrix A to matrix B, placing the result in matrix C
t	transpose matrix A
∗	multiply matrix A and B, placing the result in matrix C
p	print matrixA elements in reverse order
m	compute a minor of matrix A, placing the result in matrix C
d	compute the determinant of matrix A

In this task you only need to implement the quit command. In later tasks you will implement the remainder. Hint: make a loop in which you ask for a single character command, like this:

```
char cmd;
do {
  printf ("Command? ");
  scanf(" %c",&cmd); // notice the space before the %
  ...
} while (cmd != ’q’);
```

If an invalid character is given then print the error message `Unknown command ’Z’` (with Z replaced by the unknown command):

```
Command? X
Unknown command ’X’
Command? q
Bye!
```

Your program must produce the *exact* output, including all spaces, capitalisation, quotes, etc.

Task 2. Define three two-dimensional matrices (`matrixA`, `matrixB`, `matrixC`) in your main function of `MAXSIZE` by `MAXSIZE` `float` elements and initialise all elements to zero. You also need to define the integer variables `rowsA`, `columnsA`, `rowsB`, `columnsB`, `rowsC`, `columnsC` that indicate the size of each matrix. They are initialised to 1.
Write a `print` function that prints a `matrix` of `rows` rows and `columns` columns. It prints the floating point numbers of each matrix row on one line. The formatting of the `float` is a total of 8 positions with an accuracy of 2 positions. The `name` character is the name of the array. Use the letter 'a' for the command to print `matrixA` using this function.

```
#define MAXSIZE 10
void print (float matrix[MAXSIZE][MAXSIZE], int rows, int columns, char name) { ... }
int main (void) {
   ... print (matrixA,rowsA,columnsA,'A'); ...
}
```

A 3x3 matrix would be formatted like this:

```
Command? a
Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
```

Since the matrix `matrixA` has been initialised to a size of 1x1 with value 0 this should be the result of your program:

```
Command? a
Matrix A (1 X 1):
    0.00
Command? q
Bye!
```

Hint: Have a look in the Kernighan & Ritchie book in Appendix B1.2 to see how to format the floating point numbers (field width 8, accuracy 2).

Task 3. Next, in your main function add code to set `matrixA` to an identity matrix. Use the letter 'I' (captical letter 'i') for this command. An identity matrix is a square matrix (the number of rows equals the number of columns) in which all elements are zero, except those on the (i,i) diagonal, which are 1. Print an error message if the rows and columns are not the same, or if they are out of range.

```
Command? a
Matrix A (1 X 1):
    0.00
Command? I
Size of matrix A (rows columns)? 4 4
Command? a
Matrix A (4 X 4):
    1.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00
    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    1.00
Command? I
Size of matrix A (rows columns)? 11 11
Rows & columns must be equal and between 1 and 10
Command? I
Size of matrix A (rows columns)? 2 4
Rows & columns must be equal and between 1 and 10
Command? q
Bye!
```

Task 4. In this task you should write code in your main function to ask for the size of matrix `matrixA` and initialise `matrixA` with the floating point values entered by the user. Give an error message if the size of the matrix is out of range. Use the letter 'A' for this command.

```
Command? a
Matrix A (1 X 1):
    0.00
Command? A
Size of matrix A (rows columns)? 1 11
Rows & columns must be between 1 and 10
Command? A
Size of matrix A (rows columns)? 2 6
Row 0? 1 2 3 4 5 6
Row 1? -1 -2 -3 -4 -5 -6
Command? a
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? q
Bye!
```

Task 5. Add the 'b' command to print `matrixB` and the 'c' command to print `matrixC`, by calling the `print` function that you wrote in Task 2. Next add the 'B' command that that copies `matrixA` to `matrixB`.

```
Command? I
Size of matrix A (rows columns)? 3 3
Command? a
Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? b
Matrix B (1 X 1):
    0.00
Command? B
Command? b
Matrix B (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? A
Size of matrix A (rows columns)? 1 3
Row 0? 10 11 12
Command? a
Matrix A (1 X 3):
   10.00   11.00   12.00
Command? b
Matrix B (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? q
Bye!
```

Task 6. Add the add function to implement the '+' command to add `matrixA` to `matrixB`, placing the result in `matrixC`. If the dimensions of `matrixA` and `matrixB` are not the same then the error message `Dimensions of A & B do not match` should be printed and `matrixC` is left unchanged.

```
void add (
    float matrixA[MAXSIZE][MAXSIZE], int rowsA, int columnsA,
    float matrixB[MAXSIZE][MAXSIZE], int rowsB, int columnsB,
    float matrixC[MAXSIZE][MAXSIZE], int rowsC, int columnsC) { ...
```

Hint: Set `rowsC` and `columnsC` to the appropriate values *before* calling add.

Example output:

```
Command? A
Size of matrix A (rows columns)? 3 3
Row 0? 1 2 3
Row 1? 4 5 6
Row 2? 7 8 9
Command? B
Command? I
Size of matrix A (rows columns)? 3 3
Command? a
Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? b
Matrix B (3 X 3):
    1.00    2.00    3.00
    4.00    5.00    6.00
    7.00    8.00    9.00
Command? +
Command? c
Matrix C (3 X 3):
    2.00    2.00    3.00
    4.00    6.00    6.00
    7.00    8.00   10.00
Command? I
Size of matrix A (rows columns)? 2 2
Command? +
Dimensions of A & B do not match
Command? q
Bye!
```

Task 7. Implement the 't' command to transpose `matrixA`. To transpose a matrix each (i,j)th element is swapped with the (j,i)th element. Transposing a N*M matrix results in a M*N matrix.

```
Command? A
Size of matrix A (rows columns)? 2 3
Row 0? 1 2 3
Row 1? 4 5 6
Command? t
Command? a
Matrix A (3 X 2):
    1.00    4.00
    2.00    5.00
    3.00    6.00
Command? q
Bye!
```

Task 8. Add the `mult` function to implement the '*' command to multiply `matrixA` with `matrixB`, placing the result in `matrixC`. If the dimensions of `matrixA` and `matrixB` do not match then the error message `Dimensions of A & B do not match` should be printed and `matrixC` is left unchanged.

```
void mult (
    float matrixA[MAXSIZE][MAXSIZE], int rowsA, int columnsA,
    float matrixB[MAXSIZE][MAXSIZE], int rowsB, int columnsB,
    float matrixC[MAXSIZE][MAXSIZE], int rowsC, int columnsC) { ...
```

Recall that the formula to multiply two matrices A (rowsA X columnsA) and B (rowsB X columnsB) to result in C (rowsA X columnsB) is as follows:

$$\forall 0 \le r < \texttt{rowsA}.\ \ \forall 0 \le c < \texttt{columnsB}.\ \ C(r,c) = \sum_{0 \le i < \texttt{columnsA}} A(r,i) \times B(i,c)$$

(The $\forall$ symbol means "for all". The $\sum$ formula means adding `A(r,0)`× `B(0,c) + A(r,1)`× `B(1,c) + ... + A(r,columnsA-1)`× `B(columnsA-1,c)`.)

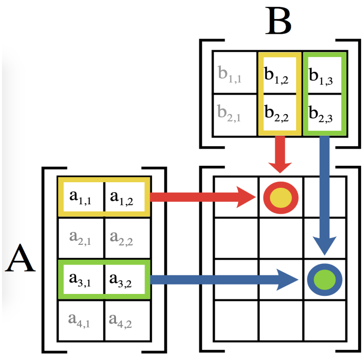


Figure 2: Matrix C is the multiplication of matrix A with matrix B.

For example, the multiplication of 1 X 5 and 5 X 1 matrices results in a 1 X 1 matrix, and the multiplication of 5 X 1 and 1 X 5 matrices results in a 5 X 5 matrix.

```
Command? A
Size of matrix A (rows columns)? 2 3
Row 0? 1 2 3
Row 1? 4 5 6
Command? B
Command? t
Command? a
Matrix A (3 X 2):
    1.00    4.00
    2.00    5.00
    3.00    6.00
Command? b
Matrix B (2 X 3):
    1.00    2.00    3.00
    4.00    5.00    6.00
Command? *
Command? c
Matrix C (3 X 3):
   17.00   22.00   27.00
   22.00   29.00   36.00
   27.00   36.00   45.00
Command? q
Bye!
```

Task 9. Add the 'p' command to print the elements of `matrixA` in reverse order. Write a function `printRecursive`:

```
void printRecursive (float matrixA[MAXSIZE][MAXSIZE],
                     int rowsA, int columnsA, int currentRow, int currentColumn)
{
  printf("enter printRecursive with current row=%d column=%d\n", currentRow, currentColumn);
  // stop recursion?
  // compute nextRow, nextColumn
  printRecursive (matrixA, rowsA, columnsA, nextRow, nextColumn);
  printf("exit printRecursive with current row=%d column=%d\n", currentRow, currentColumn);
}
int main(void)
{ ...
  case 'p':
  printRecursive(matrixA,rowsA,columnsA,0,0);
  break;
```

The function must be recursive and not use any loops. In other words, it must call itself. The first thing to decide is how to update currentRow, currentColumn before the function calls itself. Then you must decide when to not call the function again (the terminating condition) otherwise you'll end up with an infinite loop. You need to add only a few lines to the template above: to check if the recursion should stop (and then print `finished the recursion with current row=?  column=?`), and to compute nextRow and nextColumn.

```
Command? A
Size of matrix A (rows columns)? 3 3
Row 0? 1 2 3
Row 1? 4 5 6
Row 2? 7 8 9
Command? p
enter printRecursive with current row=0 column=0
enter printRecursive with current row=0 column=1
enter printRecursive with current row=0 column=2
enter printRecursive with current row=1 column=0
enter printRecursive with current row=1 column=1
enter printRecursive with current row=1 column=2
enter printRecursive with current row=2 column=0
enter printRecursive with current row=2 column=1
enter printRecursive with current row=2 column=2
enter printRecursive with current row=3 column=0
finished the recursion with current row=3 column=0
    9.00
exit printRecursive with current row=2 column=2
    8.00
exit printRecursive with current row=2 column=1
    7.00
exit printRecursive with current row=2 column=0
    6.00
exit printRecursive with current row=1 column=2
    5.00
exit printRecursive with current row=1 column=1
    4.00
exit printRecursive with current row=1 column=0
    3.00
exit printRecursive with current row=0 column=2
    2.00
exit printRecursive with current row=0 column=1
    1.00
exit printRecursive with current row=0 column=0
Command? q
Bye!
```

Task 10. Add the 'm' command to compute the minor of `matrixA` and place the result in `matrixC`. The minor `minor(A,i,j)` of matrix A is matrix A with row i and column j removed. In the figure below, the minors `minor(A,0,0)`, `minor(A,0,1)`, and `minor(A,0,2)` have been highlighted.

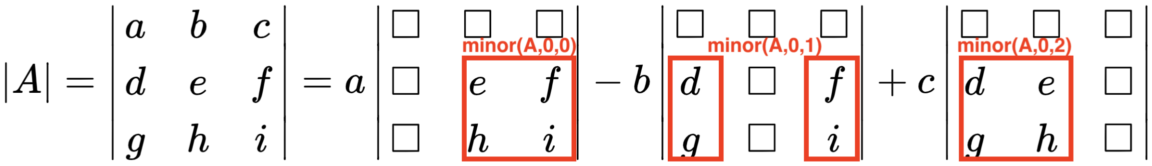


Figure 3: Minors (0,0), (0,1), and (0,2) of a 3 X 3 matrix

Write a function `minorMatrix`:

```
void minorMatrix (float matrixA[MAXSIZE][MAXSIZE], int rowsA, int columnsA,
                  int r, int c, float min[MAXSIZE][MAXSIZE]) ...
int main(void) {
  ... minorMatrix(matrixA,rowsA,columnsA,r,c,matrixC); ...
```

that removes row `r` and column `c` from `matrixA` and places the result in matrix `min`. (Set `rowsC` and `columnsC` to the right size before calling `minorMatrix`.)
If `matrixA` has fewer than 2 rows or columns or if the specified row or column is out of range then give an error message, as shown below.

```
Command? A
Size of matrix A (rows columns)? 4 3
Row 0? 3 2 1
Row 1? -3 -5 -7
Row 2? 10 20 30
Row 3? -10 -11 -12
Command? m
Remove which row & column of matrix A? 0 0
Command? c
Matrix C (3 X 2):
   -5.00   -7.00
   20.00   30.00
  -11.00  -12.00
Command? m
Remove which row & column of matrix A? 1 2
Command? c
Matrix C (3 X 2):
    3.00    2.00
   10.00   20.00
  -10.00  -11.00
Command? m
Remove which row & column of matrix A? 0 5
Rows & columns must be between 0 and 3 & 2, respectively
Command? I
Size of matrix A (rows columns)? 1 1
Command? m
Remove which row & column of matrix A? 0 0
Matrix A must have at least two rows & columns
Command? q
Bye!
```

Task 11. The final optional task is to compute the determinant of matrixA using the Laplace expansion. This is a recursive way of computing a determinant that uses the `minor` function of the previous task.
The determinant of a 1 X 1 matrix A is equal to A(0,0). The determinant of a 2 X 2 matrix A is equal to A(0,0)*A(1,1) - A(0,1)*A(1,0). The determinant of a 3 X 3 matrix A is computed by taking row zero and multiplying the elements (r,c) of the row by the minors `minor(A,r,c)`, as illustrated in the figure below. The only minor complication is that the sign of each product is either +1 or -1, as defined by $(-1)^c$. (You can use any row or column, but we recommend using row 0, as shown in the formula and figure.)

$$\texttt{determinant}(A) = \sum_{0 \leq c < \texttt{columnsA}} (-1)^c \times \texttt{determinant}(\texttt{minor(A,0,c)})$$

The formula above allows you to compute the determinant of any N X N matrix.

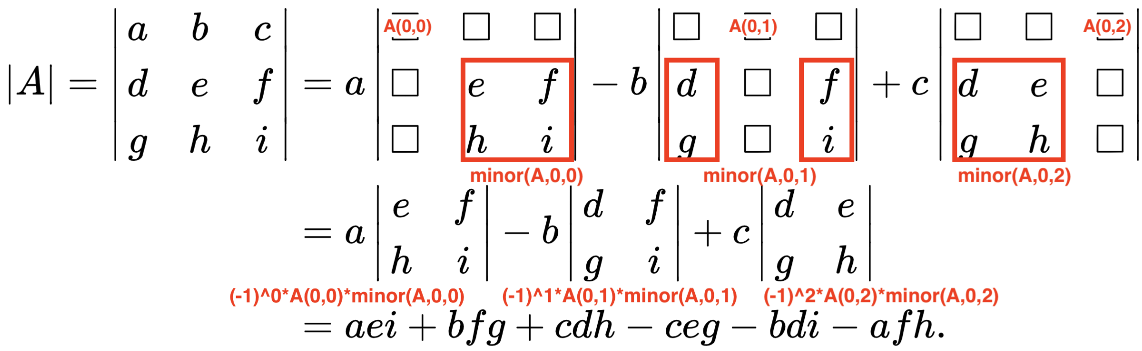

A(0,0)
A(0,1)
A(0,2)
minor(A,0,0)
minor(A,0,1)
minor(A,0,2)
(-1)^0*A(0,0)*minor(A,0,0)
(-1)^1*A(0,1)*minor(A,0,1)
(-1)^2*A(0,2)*minor(A,0,2)
= aei + bfg + cdh − ceg − bdi − afh.

Figure 4: The minors of a 3 X 3 matrix

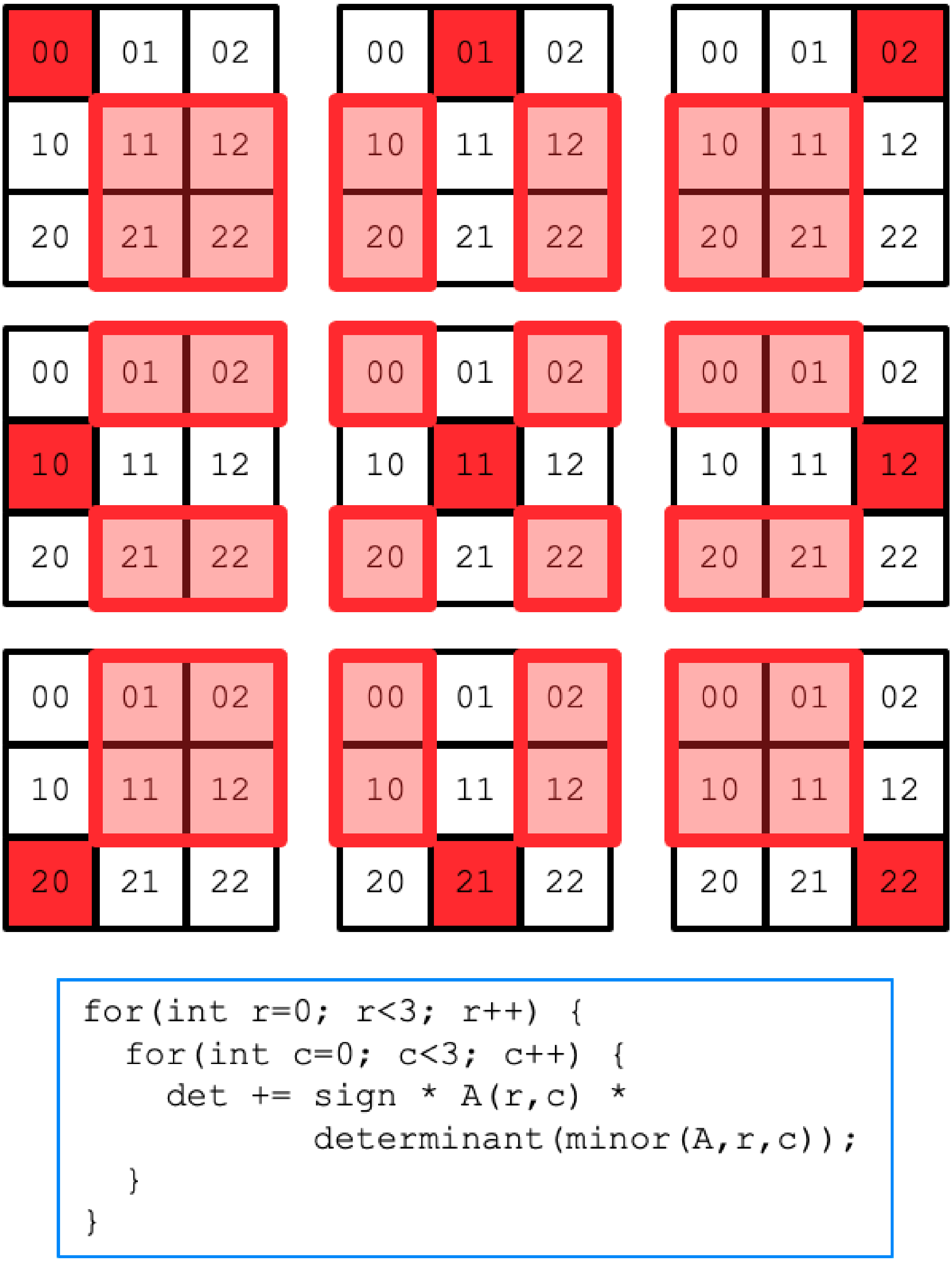
You can use the following code as a template.

```
float determinant (float matrixA[MAXSIZE][MAXSIZE], int rowsA, int columnsA)
{
  float det;
  if (rowsA == 1 && columnsA == 1) det = ...;
  else if (rowsA == 2 && columnsA == 2) det = ...;
  else {
    // recursively call determinant on the columnsA minors of A along row 0
    int sign = (c % 2 ? -1 : +1); // the sign for minor(matrixA,0,c)
  }
  printf("%d",rowsA);
  return det;
}
```

Have a look at the figures below that show how the recursion works.
If `matrixA` is not square then then give the error message `Matrix A must be square`
The `printf` in the template code is required and it shows the recursion, e.g. 2223 for a 3 X 3 matrix shows that determinant has been called three times on a 2 X 2 matrix, and once on a 3 X 3 matrix. For a 4 X 4 matrix the recursion becomes clear as determinant is called four times on a 3 X 3 matrix: 22232223222322234.

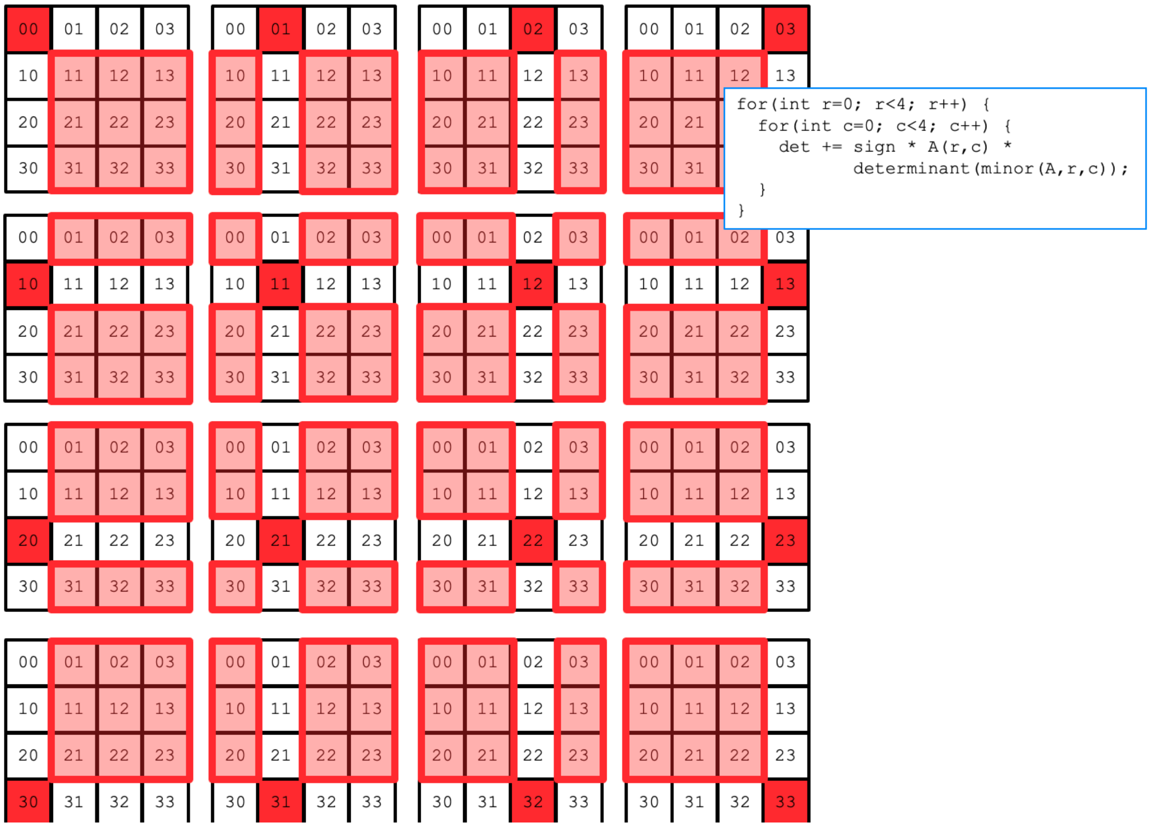
```
Command? I
Size of matrix A (rows columns)? 3 3
Command? d
2223
The determinant is 1.000000
Command? I
Size of matrix A (rows columns)? 4 4
Command? d
22232223222322234
The determinant is 1.000000
Command?
```

```
Command? A
Size of matrix A (rows columns)? 1 1
Row 0? 9
Command? d
1
The determinant is 9.000000
Command? A
Size of matrix A (rows columns)? 2 2
Row 0? 1 2
Row 1? 10 11
Command? d
2
The determinant is -9.000000
Command? A
Size of matrix A (rows columns)? 3 3
Row 0? -2 2 -3
Row 1? -1 1 3
Row 2? 2 0 -1
Command? d
2223
The determinant is 18.000000
Command? A
Size of matrix A (rows columns)? 4 4
Row 0? 1 2 -4 -5
Row 1? 2 7 3 -2
Row 2? 1 1 -1 -1
Row 3? 4 7 8 -5
Command? d
22232223222322234
The determinant is -200.000000
```



```
for(int r=0; r<3; r++) {
  for(int c=0; c<3; c++) {
    det += sign * A(r,c) *
              determinant(minor(A,r,c));
  }
}
```

Figure 5: Recursion of determinant calculation for 3x3 matrix.



```
for(int r=0; r<4; r++) {
  for(int c=0; c<4; c++) {
    det += sign * A(r,c) *
           determinant(minor(A,r,c));
  }
}
```

Figure 6: Recursion of determinant calculation for 4x4 matrix; top-level loop only.

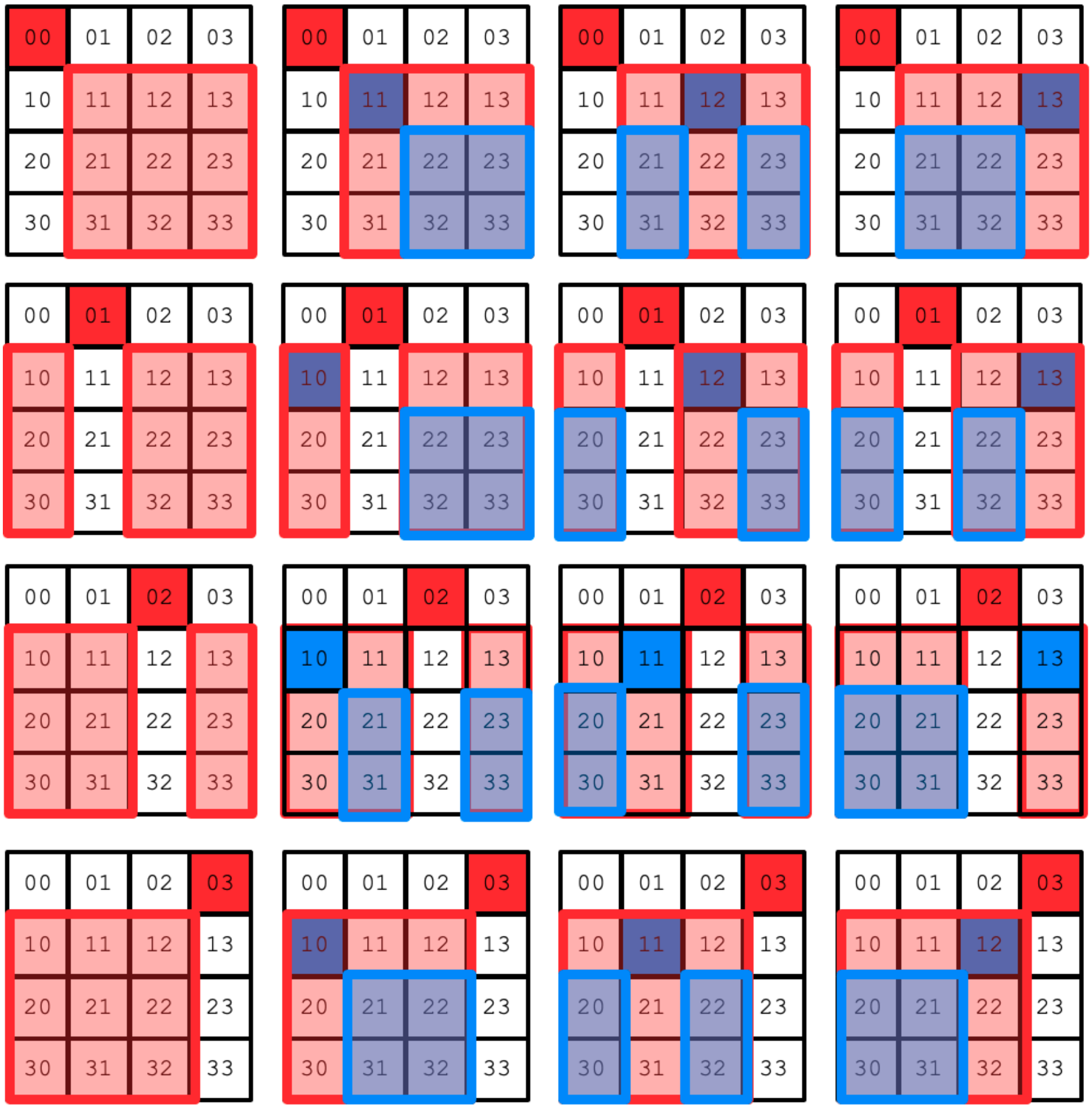


Figure 7: Recursion of determinant calculation for 4x4 matrix. Recursion for is shown for the first row only.

Submission: Your *last* submission will be automatically graded.

- **29/9 v1.1** Minor clarifications.
- **30/9 v1.2** Fixed typo.
- **29/7 v2.0** Made into homework.
- **3/8 v2.1** Added printRecursive and made determinant optional.
- **3/8 v2.2** More figures for recursion.
- **18/9 v2.4** Updated formatting to remove trailing space.

Input / output test cases

Long lines have been wrapped at 70 characters for legibility. When your program output is compared to the expected output lines will not be wrapped.

Case 01

Input:

```
q
```

Output:

```
Command? Bye!
```

Case 02

Input:

```
a
q
```

Output:

```
Command? Matrix A (1 X 1):
    0.00
Command? Bye!
```

Case 03

Input:

```
I
4 4
a
I
4 5
a
I
0 1
a
I
11 11
a
I
3 3
a
q
```

Output:

```
Command? Size of matrix A (rows columns)? Command? Matrix A (4 X 4):
    1.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00
    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    1.00
Command? Size of matrix A (rows columns)? Rows & columns must be
equal and between 1 and 10
Command? Matrix A (4 X 4):
    1.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00
    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    1.00
Command? Size of matrix A (rows columns)? Rows & columns must be
equal and between 1 and 10
Command? Matrix A (4 X 4):
    1.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00
    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    1.00
Command? Size of matrix A (rows columns)? Rows & columns must be
equal and between 1 and 10
Command? Matrix A (4 X 4):
    1.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00
    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    1.00
Command? Size of matrix A (rows columns)? Command? Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? Bye!
```

Case 04

Input:

```
a
A
1 1
-1
a
A
2 2
1 2
3 4
a
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
a
A
6 2
9 8
8 7
7 6
6 5
5 4
4 3
a
A
1 10
9 8 7 6 5 4 3 2 1 0
a
A
10 1
9
8
7
6
5
4
3
2
1
0
a
A
1 1
-1.2345
a
q
```

Output:

```
Command? Matrix A (1 X 1):
    0.00
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A (1
X 1):
   -1.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 2):
    1.00    2.00
    3.00    4.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Command? Matrix A (6 X 2):
    9.00    8.00
    8.00    7.00
    7.00    6.00
    6.00    5.00
    5.00    4.00
    4.00    3.00
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A (1
X 10):
    9.00    8.00    7.00    6.00    5.00    4.00    3.00    2.00
1.00    0.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Row 6? Row 7? Row 8? Row 9? Command? Matrix A (10 X 1):
    9.00
    8.00
    7.00
    6.00
    5.00
    4.00
    3.00
    2.00
    1.00
    0.00
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A (1
X 1):
   -1.23
Command? Bye!
```

Case 05

Input:

```
I
0 0
a
A
0 1
A
11 0
a
b
c
q
```

Output:

```
Command? Size of matrix A (rows columns)? Rows & columns must be
equal and between 1 and 10
Command? Matrix A (1 X 1):
    0.00
Command? Size of matrix A (rows columns)? Rows & columns must be
between 1 and 10
Command? Size of matrix A (rows columns)? Rows & columns must be
between 1 and 10
Command? Matrix A (1 X 1):
    0.00
Command? Matrix B (1 X 1):
    0.00
Command? Matrix C (1 X 1):
    0.00
Command? Bye!
```

Case 06

Input:

```
I
5 5
a
b
c
B
a
b
c
I
2 2
a
b
c
B
a
b
c
q
```

Output:

```
Command? Size of matrix A (rows columns)? Command? Matrix A (5 X 5):
    1.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    1.00
Command? Matrix B (1 X 1):
    0.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Matrix A (5 X 5):
    1.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    1.00
Command? Matrix B (5 X 5):
    1.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    1.00
Command? Matrix C (1 X 1):
    0.00
Command? Size of matrix A (rows columns)? Command? Matrix A (2 X 2):
    1.00    0.00
    0.00    1.00
Command? Matrix B (5 X 5):
    1.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    1.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Matrix A (2 X 2):
    1.00    0.00
    0.00    1.00
Command? Matrix B (2 X 2):
    1.00    0.00
    0.00    1.00
Command? Matrix C (1 X 1):
    0.00
Command? Bye!
```

Case 07

Input:

```
a
b
c
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
a
b
c
B
a
b
c
A
6 2
9 8
8 7
7 6
6 5
5 4
4 3
a
b
c
B
a
b
c
q
```

Output:

```
Command? Matrix A (1 X 1):
    0.00
Command? Matrix B (1 X 1):
    0.00
Command? Matrix C (1 X 1):
    0.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix B (1 X 1):
    0.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix B (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix C (1 X 1):
    0.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Command? Matrix A (6 X 2):
    9.00    8.00
    8.00    7.00
    7.00    6.00
    6.00    5.00
    5.00    4.00
    4.00    3.00
Command? Matrix B (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Matrix A (6 X 2):
    9.00    8.00
    8.00    7.00
    7.00    6.00
    6.00    5.00
    5.00    4.00
    4.00    3.00
Command? Matrix B (6 X 2):
    9.00    8.00
    8.00    7.00
    7.00    6.00
    6.00    5.00
    5.00    4.00
    4.00    3.00
Command? Matrix C (1 X 1):
    0.00
Command? Bye!
```

Case 08

Input:

```
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
+
a
b
c
B
+
a
b
c
I
7 7
B
+
a
b
c
+
a
b
c
A
7 7
1 2 3 4 5 6 7
11 -12 13 14 15 16 17
1 2 3 -4 -5 6 7
-1 -2 3 4 5 6 7
11 12 13 -14 15 16 17
1 2 3 4 5 -6 -7
11 12 -13 14 15 16 17
a
b
c
+
a
b
c
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Dimensions of A & B do not match
Command? Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix B (1 X 1):
    0.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Command? Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix B (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Matrix C (2 X 6):
    2.00    4.00    6.00    8.00   10.00   12.00
   -2.00   -4.00   -6.00   -8.00  -10.00  -12.00
Command? Size of matrix A (rows columns)? Command? Command? Command?
Matrix A (7 X 7):
    1.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    0.00    0.00    1.00
Command? Matrix B (7 X 7):
    1.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    0.00    0.00    1.00
Command? Matrix C (7 X 7):
    2.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    2.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    2.00    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    2.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00    2.00    0.00    0.00
    0.00    0.00    0.00    0.00    0.00    2.00    0.00
    0.00    0.00    0.00    0.00    0.00    0.00    2.00
Command? Command? Matrix A (7 X 7):
    1.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    0.00    0.00    1.00
Command? Matrix B (7 X 7):
    1.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    1.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    1.00    0.00    0.00    0.00    0.00
    0.00    0.00    0.00    1.00    0.00    0.00    0.00
    0.00    0.00    0.00    0.00    1.00    0.00    0.00
    0.00    0.00    0.00    0.00    0.00    1.00    0.00
    0.00    0.00    0.00    0.00    0.00    0.00    1.00
Command? Matrix C (7 X 7):
    2.00    0.00    0.00    0.00    0.00    0.00    0.00
    0.00    2.00    0.00    0.00    0.00    0.00    0.00
    0.00    0.00    2.00    0.00    0.00    0.00    0.00
```

Case 09

Input:

```
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
I
7 7
+
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command? Size
of matrix A (rows columns)? Command? Dimensions of A & B do not match
Command? Bye!
```

Case 10

Input:

```
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
a
A
2 2
1 2
11 -12
a
A
1 1
3
B
+
a
b
c
A
2 1
3
4
B
+
a
b
c
A
1 3
3 2 1
B
+
a
b
c
A
10 10
1 2 3 4 5 6 7 8 9 10
1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 10
2 2 2 2 2 2 2 2 2 2
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 10
2 2 2 2 2 2 2 2 2 2
1 2 3 4 5 6 7 8 9 10
a
B
I
10 10
+
c
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 2):
    1.00    2.00
   11.00  -12.00
Command? Size of matrix A (rows columns)? Row 0? Command? Command?
Command? Matrix A (1 X 1):
    3.00
Command? Matrix B (1 X 1):
    3.00
Command? Matrix C (1 X 1):
    6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Command? Command? Matrix A (2 X 1):
    3.00
    4.00
Command? Matrix B (2 X 1):
    3.00
    4.00
Command? Matrix C (2 X 1):
    6.00
    8.00
Command? Size of matrix A (rows columns)? Row 0? Command? Command?
Command? Matrix A (1 X 3):
    3.00    2.00    1.00
Command? Matrix B (1 X 3):
    3.00    2.00    1.00
Command? Matrix C (1 X 3):
    6.00    4.00    2.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Row 6? Row 7? Row 8? Row 9? Command? Matrix A (10 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
1.00    1.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    2.00    2.00    2.00    2.00    2.00    2.00    2.00    2.00
2.00    2.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
1.00    1.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    2.00    2.00    2.00    2.00    2.00    2.00    2.00    2.00
2.00    2.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
Command? Command? Size of matrix A (rows columns)? Command? Command?
Matrix C (10 X 10):
    2.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    1.00    1.00    1.00    1.00    1.00    1.00
1.00    1.00
    1.00    2.00    4.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    2.00    2.00    2.00    3.00    2.00    2.00    2.00    2.00
```

Case 11

Input:

```
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
a
t
a
A
2 2
1 2
11 -12
a
t
a
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Command? Matrix A (6 X 2):
    1.00   -1.00
    2.00   -2.00
    3.00   -3.00
    4.00   -4.00
    5.00   -5.00
    6.00   -6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 2):
    1.00    2.00
   11.00  -12.00
Command? Command? Matrix A (2 X 2):
    1.00   11.00
    2.00  -12.00
Command? Bye!
```

Case 12

Input:

```
A
2 6
1 2 3 4 5 6
-1 -2 -3 -4 -5 -6
a
t
a
A
7 7
1 2 3 4 5 6 7
11 -12 13 14 15 16 17
1 2 3 -4 -5 6 7
-1 -2 3 4 5 6 7
11 12 13 -14 15 16 17
1 2 3 4 5 -6 -7
11 12 -13 14 15 16 17
a
t
a
I
3 3
a
t
a
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
   -1.00   -2.00   -3.00   -4.00   -5.00   -6.00
Command? Command? Matrix A (6 X 2):
    1.00   -1.00
    2.00   -2.00
    3.00   -3.00
    4.00   -4.00
    5.00   -5.00
    6.00   -6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Row 6? Command? Matrix A (7 X 7):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00
   11.00  -12.00   13.00   14.00   15.00   16.00   17.00
    1.00    2.00    3.00   -4.00   -5.00    6.00    7.00
   -1.00   -2.00    3.00    4.00    5.00    6.00    7.00
   11.00   12.00   13.00  -14.00   15.00   16.00   17.00
    1.00    2.00    3.00    4.00    5.00   -6.00   -7.00
   11.00   12.00  -13.00   14.00   15.00   16.00   17.00
Command? Command? Matrix A (7 X 7):
    1.00   11.00    1.00   -1.00   11.00    1.00   11.00
    2.00  -12.00    2.00   -2.00   12.00    2.00   12.00
    3.00   13.00    3.00    3.00   13.00    3.00  -13.00
    4.00   14.00   -4.00    4.00  -14.00    4.00   14.00
    5.00   15.00   -5.00    5.00   15.00    5.00   15.00
    6.00   16.00    6.00    6.00   16.00   -6.00   16.00
    7.00   17.00    7.00    7.00   17.00   -7.00   17.00
Command? Size of matrix A (rows columns)? Command? Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? Command? Matrix A (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? Bye!
```

Case 13

Input:

```
A
6 1
1
2
3
4
5
6
a
t
a
t
a
t
t
t
t
a
A
3 10
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
B
a
b
+
b
t
a
b
A
7 7
1 2 3 4 5 6 7
11 -12 13 14 15 16 17
1 2 3 -4 -5 6 7
-1 -2 3 4 5 6 7
11 12 13 -14 15 16 17
1 2 3 4 5 -6 -7
11 12 -13 14 15 16 17
a
t
a
I
3 3
a
t
t
t
t
t
t
t
t
t
t
t
t
t
t
t
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Command? Matrix A (6 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
    6.00
Command? Command? Matrix A (1 X 6):
    1.00    2.00    3.00    4.00    5.00    6.00
Command? Command? Matrix A (6 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
    6.00
Command? Command? Command? Command? Command? Matrix A (6 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
    6.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2?
Command? Command? Matrix A (3 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
   11.00   12.00   13.00   14.00   15.00   16.00   17.00   18.00
19.00   20.00
   21.00   22.00   23.00   24.00   25.00   26.00   27.00   28.00
29.00   30.00
Command? Matrix B (3 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
   11.00   12.00   13.00   14.00   15.00   16.00   17.00   18.00
19.00   20.00
   21.00   22.00   23.00   24.00   25.00   26.00   27.00   28.00
29.00   30.00
Command? Command? Matrix B (3 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
   11.00   12.00   13.00   14.00   15.00   16.00   17.00   18.00
19.00   20.00
   21.00   22.00   23.00   24.00   25.00   26.00   27.00   28.00
29.00   30.00
Command? Command? Matrix A (10 X 3):
    1.00   11.00   21.00
    2.00   12.00   22.00
    3.00   13.00   23.00
    4.00   14.00   24.00
    5.00   15.00   25.00
    6.00   16.00   26.00
    7.00   17.00   27.00
    8.00   18.00   28.00
    9.00   19.00   29.00
   10.00   20.00   30.00
Command? Matrix B (3 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
   11.00   12.00   13.00   14.00   15.00   16.00   17.00   18.00
19.00   20.00
   21.00   22.00   23.00   24.00   25.00   26.00   27.00   28.00
```

Case 14

Input:

```
A
3 1
1
2
3
B
A
1 3
-1 -2 -3
a
b
c
*
a
b
c
A
3 2
1 2
-1 -2
1 1
B
t
*
a
b
c
q
```

Output:

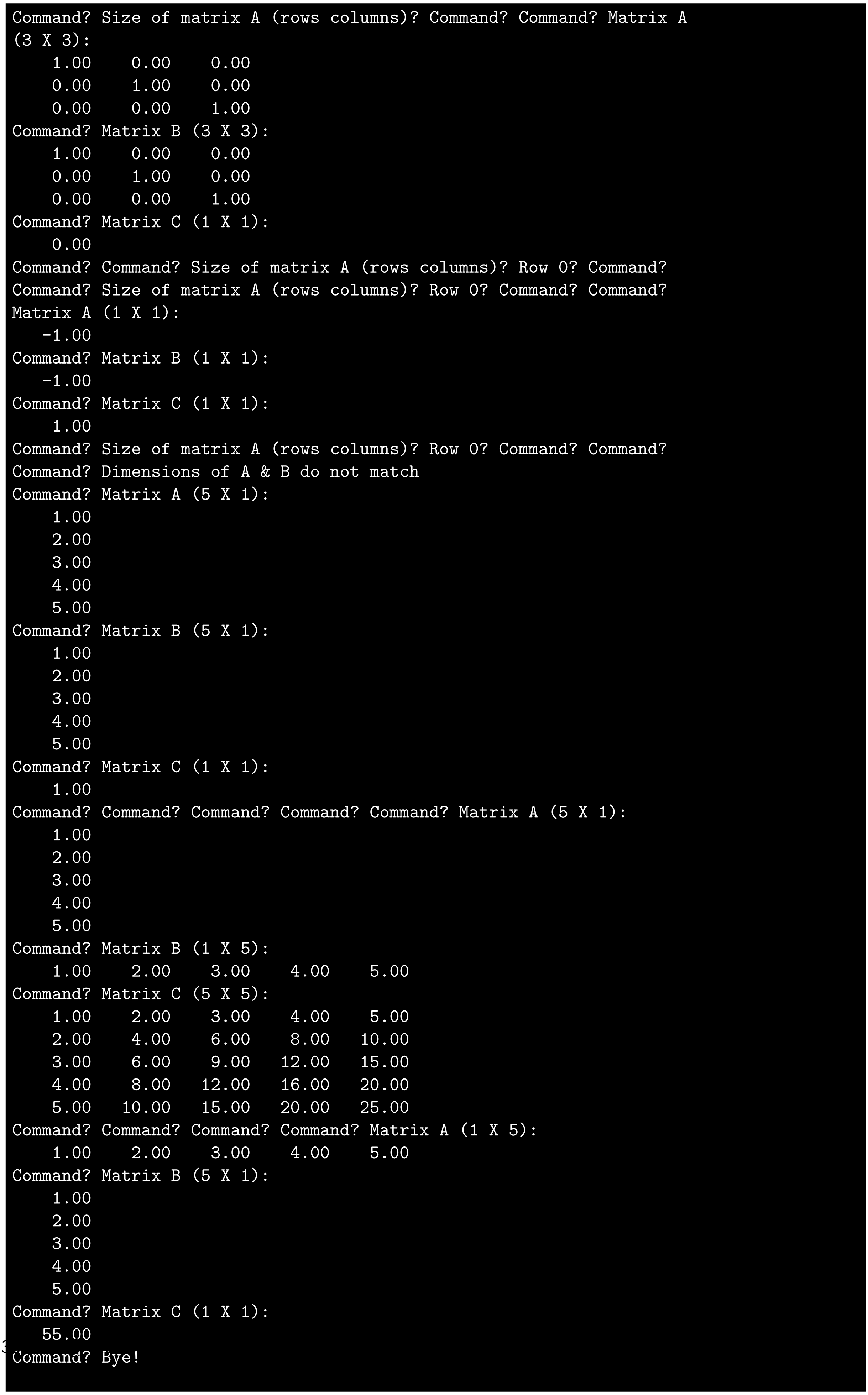
```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2?
Command? Command? Size of matrix A (rows columns)? Row 0? Command?
Matrix A (1 X 3):
   -1.00   -2.00   -3.00
Command? Matrix B (3 X 1):
    1.00
    2.00
    3.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Matrix A (1 X 3):
   -1.00   -2.00   -3.00
Command? Matrix B (3 X 1):
    1.00
    2.00
    3.00
Command? Matrix C (1 X 1):
  -14.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2?
Command? Command? Command? Command? Matrix A (2 X 3):
    1.00   -1.00    1.00
    2.00   -2.00    1.00
Command? Matrix B (3 X 2):
    1.00    2.00
   -1.00   -2.00
    1.00    1.00
Command? Matrix C (2 X 2):
    3.00    5.00
    5.00    9.00
Command? Bye!
```

Case 15

Input:

```
I
3 3
B
a
b
c
*
A
1 1
-1
B
A
1 1
-1
*
a
b
c
A 1 5
1 2 3 4 5
t
B
*
a
b
c
t
B
t
*
a
b
c
B
t
*
a
b
c
q
```

Output:



```
Command? Size of matrix A (rows columns)? Command? Command? Matrix A
(3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? Matrix B (3 X 3):
    1.00    0.00    0.00
    0.00    1.00    0.00
    0.00    0.00    1.00
Command? Matrix C (1 X 1):
    0.00
Command? Command? Size of matrix A (rows columns)? Row 0? Command?
Command? Size of matrix A (rows columns)? Row 0? Command? Command?
Matrix A (1 X 1):
   -1.00
Command? Matrix B (1 X 1):
   -1.00
Command? Matrix C (1 X 1):
    1.00
Command? Size of matrix A (rows columns)? Row 0? Command? Command?
Command? Dimensions of A & B do not match
Command? Matrix A (5 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
Command? Matrix B (5 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
Command? Matrix C (1 X 1):
    1.00
Command? Command? Command? Command? Command? Matrix A (5 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
Command? Matrix B (1 X 5):
    1.00    2.00    3.00    4.00    5.00
Command? Matrix C (5 X 5):
    1.00    2.00    3.00    4.00    5.00
    2.00    4.00    6.00    8.00   10.00
    3.00    6.00    9.00   12.00   15.00
    4.00    8.00   12.00   16.00   20.00
    5.00   10.00   15.00   20.00   25.00
Command? Command? Command? Command? Matrix A (1 X 5):
    1.00    2.00    3.00    4.00    5.00
Command? Matrix B (5 X 1):
    1.00
    2.00
    3.00
    4.00
    5.00
Command? Matrix C (1 X 1):
   55.00
Command? Bye!
```

Case 16

Input:

```
A
10 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
B
*
a
b
c
+
a
b
c
A
1 10
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
*
c
A
2 10
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-2 -2 -2 -2 -2 -2 -2 -2 -2 -2
*
c
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Row 6? Row 7? Row 8? Row 9? Command? Command? Command?
Matrix A (10 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
Command? Matrix B (10 X 10):
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
    1.00    2.00    3.00    4.00    5.00    6.00    7.00    8.00
9.00   10.00
Command? Matrix C (10 X 10):
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
   55.00  110.00  165.00  220.00  275.00  330.00  385.00  440.00
495.00  550.00
```

Case 17

Input:

```
p
A
2 2
1 2
3 4
a
p
A
4 3
3 2 1
-3 -5 -7
1 2 3
-10 -11 -12
a
p
q
```

Output:

```
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=1 column=0
finished the recursion with current row=1 column=0
    0.00
exit printRecursive with current row=0 column=0
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 2):
    1.00    2.00
    3.00    4.00
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=0 column=1
enter printRecursive with current row=1 column=0
enter printRecursive with current row=1 column=1
enter printRecursive with current row=2 column=0
finished the recursion with current row=2 column=0
    4.00
exit printRecursive with current row=1 column=1
    3.00
exit printRecursive with current row=1 column=0
    2.00
exit printRecursive with current row=0 column=1
    1.00
exit printRecursive with current row=0 column=0
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Command? Matrix A (4 X 3):
    3.00    2.00    1.00
   -3.00   -5.00   -7.00
    1.00    2.00    3.00
  -10.00  -11.00  -12.00
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=0 column=1
enter printRecursive with current row=0 column=2
enter printRecursive with current row=1 column=0
enter printRecursive with current row=1 column=1
enter printRecursive with current row=1 column=2
enter printRecursive with current row=2 column=0
enter printRecursive with current row=2 column=1
enter printRecursive with current row=2 column=2
enter printRecursive with current row=3 column=0
enter printRecursive with current row=3 column=1
enter printRecursive with current row=3 column=2
enter printRecursive with current row=4 column=0
finished the recursion with current row=4 column=0
  -12.00
exit printRecursive with current row=3 column=2
  -11.00
exit printRecursive with current row=3 column=1
  -10.00
exit printRecursive with current row=3 column=0
    3.00
exit printRecursive with current row=2 column=2
    2.00
exit printRecursive with current row=2 column=1
    1.00
exit printRecursive with current row=2 column=0
   -7.00
exit printRecursive with current row=1 column=2
   -5.00
exit printRecursive with current row=1 column=1
   -3.00
exit printRecursive with current row=1 column=0
    1.00
exit printRecursive with current row=0 column=2
```

Case 18

Input:

```
p
A
1 2
1 2
a
p
A
2 1
1
2
a
p
q
```

Output:

```
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=1 column=0
finished the recursion with current row=1 column=0
    0.00
exit printRecursive with current row=0 column=0
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A (1
X 2):
    1.00    2.00
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=0 column=1
enter printRecursive with current row=1 column=0
finished the recursion with current row=1 column=0
    2.00
exit printRecursive with current row=0 column=1
    1.00
exit printRecursive with current row=0 column=0
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 1):
    1.00
    2.00
Command? enter printRecursive with current row=0 column=0
enter printRecursive with current row=1 column=0
enter printRecursive with current row=2 column=0
finished the recursion with current row=2 column=0
    2.00
exit printRecursive with current row=1 column=0
    1.00
exit printRecursive with current row=0 column=0
Command? Bye!
```

Case 19

Input:

```
A
4 3
3 2 1
-3 -5 -7
1 2 3
-10 -11 -12
a
m
0 0
c
m
0 1
c
m
0 2
c
m
1 0
c
m
1 1
c
m
1 2
c
A
1 1
9
m
0 0
A
1 2
1 1
m
0 0
A
2 1
1
1
m
0 0
A
8 2
1 2
3 4
5 6
7 8
1 2
3 4
5 6
7 8
a
m
0 0
c
m
7 0
c
m
7 1
c
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Command? Matrix A (4 X 3):
    3.00    2.00    1.00
   -3.00   -5.00   -7.00
    1.00    2.00    3.00
  -10.00  -11.00  -12.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
   -5.00   -7.00
    2.00    3.00
  -11.00  -12.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
   -3.00   -7.00
    1.00    3.00
  -10.00  -12.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
   -3.00   -5.00
    1.00    2.00
  -10.00  -11.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
    2.00    1.00
    2.00    3.00
  -11.00  -12.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
    3.00    1.00
    1.00    3.00
  -10.00  -12.00
Command? Remove which row & column of matrix A? Command? Matrix C (3
X 2):
    3.00    2.00
    1.00    2.00
  -10.00  -11.00
Command? Size of matrix A (rows columns)? Row 0? Command? Remove
which row & column of matrix A? Matrix A must have at least two rows
& columns
Command? Size of matrix A (rows columns)? Row 0? Command? Remove
which row & column of matrix A? Matrix A must have at least two rows
& columns
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Remove which row & column of matrix A? Matrix A must have at least
two rows & columns
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Row 4? Row 5? Row 6? Row 7? Command? Matrix A (8 X 2):
    1.00    2.00
    3.00    4.00
    5.00    6.00
    7.00    8.00
    1.00    2.00
    3.00    4.00
    5.00    6.00
    7.00    8.00
Command? Remove which row & column of matrix A? Command? Matrix C (7
X 1):
    4.00
    6.00
    8.00
    2.00
    4.00
    6.00
```

Case 20

Input:

```
A
1 2
1 2
d
A
1 1
9
a
d
A
2 2
1 2
3 -4
a
d
A
3 3
1 2 3
1 1 1
-1 -1 -1
a
t
d
m
0 0
c
m
1 1
c
m
2 2
c
A
4 4
0.5 0.3 0.4 0.5
1 8 6 3
-4 5 -90 3
0.01 0.02 2 3
a
d
B
+
a
b
c
t
*
a
b
c
I
7 7
d
q
```

Output:

```
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A
must be square
Command? Size of matrix A (rows columns)? Row 0? Command? Matrix A (1
X 1):
    9.00
Command? 1
The determinant is 9.000000
Command? Size of matrix A (rows columns)? Row 0? Row 1? Command?
Matrix A (2 X 2):
    1.00    2.00
    3.00   -4.00
Command? 2
The determinant is -10.000000
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2?
Command? Matrix A (3 X 3):
    1.00    2.00    3.00
    1.00    1.00    1.00
   -1.00   -1.00   -1.00
Command? Command? 2223
The determinant is 0.000000
Command? Remove which row & column of matrix A? Command? Matrix C (2
X 2):
    1.00   -1.00
    1.00   -1.00
Command? Remove which row & column of matrix A? Command? Matrix C (2
X 2):
    1.00   -1.00
    3.00   -1.00
Command? Remove which row & column of matrix A? Command? Matrix C (2
X 2):
    1.00    1.00
    2.00    1.00
Command? Size of matrix A (rows columns)? Row 0? Row 1? Row 2? Row 3?
Command? Matrix A (4 X 4):
    0.50    0.30    0.40    0.50
    1.00    8.00    6.00    3.00
   -4.00    5.00  -90.00    3.00
    0.01    0.02    2.00    3.00
Command? 22232223222322234
The determinant is -1053.177979
Command? Command? Command? Matrix A (4 X 4):
    0.50    0.30    0.40    0.50
    1.00    8.00    6.00    3.00
   -4.00    5.00  -90.00    3.00
    0.01    0.02    2.00    3.00
Command? Matrix B (4 X 4):
    0.50    0.30    0.40    0.50
    1.00    8.00    6.00    3.00
   -4.00    5.00  -90.00    3.00
    0.01    0.02    2.00    3.00
Command? Matrix C (4 X 4):
    1.00    0.60    0.80    1.00
    2.00   16.00   12.00    6.00
   -8.00   10.00 -180.00    6.00
    0.02    0.04    4.00    6.00
Command? Command? Command? Matrix A (4 X 4):
    0.50    1.00   -4.00    0.01
    0.30    8.00    5.00    0.02
    0.40    6.00  -90.00    2.00
    0.50    3.00    3.00    3.00
Command? Matrix B (4 X 4):
    0.50    0.30    0.40    0.50
    1.00    8.00    6.00    3.00
```