Sample Solution to Assignment 1, Problem 3

COURSE HOME
SYLLABUS
CALENDAR
GETTING STARTED
LECTURE NOTES
ASSIGNMENTS
RELATED RESOURCES
DOWNLOAD COURSE MATERIALS

« Back to Assignments

```
/*

PROG: matrix
LANG: C
*/
#include <stdio.h>
#include <stdlib.h>
#define MAXN 300

typedef struct Matrix {
  size_t R, C;
  int index[MAXN][MAXN];
} Matrix;

void read_matrix( FILE *fin, Matrix *matrix ) {
  fscanf( fin, "%zu %zu", &matrix->R, &matrix->C );

  if( matrix->R >= MAXN || matrix->C >= MAXN ) {
    printf( "Error: tried to read matrix with a dimension larger than %d\n", MAXN );
    exit( EXIT_FAILURE );
  }

  for( size_t r = 0; r < matrix->R; ++r ) {
    for( size_t c = 0; c < matrix->C; ++c ) {
      fscanf( fin, "%d", &matrix->index[r][c] );
    }
  }
}

void print_matrix( FILE *fout, Matrix *matrix ) {
  fprintf( fout, "%zu %zu\n", matrix->R, matrix->C );
  for( size_t r = 0; r < matrix->R; ++r ) {
```

```
    for( size_t c = 0; c < matrix->C - 1; ++c ) {
      fprintf( fout, "%d ", matrix->index[r][c] );
    }
    fprintf( fout, "%d\n", matrix->index[r][matrix->C - 1] );
  }
}

void mult_matrix( Matrix *a, Matrix *b, Matrix *prod ) {
  if( a->C != b->R ) {
    printf( "Error: tried to multiply (%zux%zu)x(%zux%zu)\n", a->R, a->C, b->R, b->C );
    exit( EXIT_FAILURE );
  }

  size_t inner = a->C;
  prod->R = a->R;
  prod->C = b->C;

  for( size_t r = 0; r < prod->R; ++r ) {
    for( size_t c = 0; c < prod->C; ++c ) {
      prod->index[r][c] = 0;
      for( size_t i = 0; i < inner; ++i ) {
        prod->index[r][c] += a->index[r][i] * b->index[i][c];
      }
    }
  }
}

int main(void) {
  FILE *fin = fopen( "matrix.in", "r" ),
       *fout = fopen( "matrix.out", "w" );

  if( fin == NULL ) {
    printf( "Error: could not open matrix.in\n" );
    exit( EXIT_FAILURE );
  }

  if( fin == NULL ) {
    printf( "Error: could not open matrix.out\n" );
    exit( EXIT_FAILURE );
  }

  Matrix a, b, c;

  read_matrix( fin, &a );
  read_matrix( fin, &b );
  fclose( fin );

  mult_matrix( &a, &b, &c );
```

```
  print_matrix( fout, &c );
  fclose( fout );

  return 0;
}
```

Below is the output using the test data:

matrix:

```
 1: OK [0.004 seconds]
 2: OK [0.004 seconds]
 3: OK [0.004 seconds]
 4: OK [0.013 seconds]
 5: OK [0.009 seconds]
 6: OK [0.006 seconds]
 7: OK [0.011 seconds]
 8: OK [0.011 seconds]
 9: OK [0.012 seconds]
10: OK [0.004 seconds]
```

« *Back to Assignments*

ABOUT MIT OPENCOURSEWARE

MIT OpenCourseWare makes the materials used in the teaching of almost all of MIT's subjects available on the Web, free of charge. With more than 2,400 courses available, OCW is delivering on the promise of open sharing of knowledge. Learn more »