

```

/* =====
 *
 * CS 566 - Assignment 03
 * Camillo Lugaresi, Cosmin Stroe
 *
 * Header file for common.c
 *
 * ===== */

/* CELL(matrix,r,c) - for matrices in row,column order */
#define CELL(m,r,c) (((m)->data)[((m)->n)*(r) + (c)])

/* BLK_CELL */
#define BLK_CELL(data,blksz,b,r,c) ((data)[(blksz*blksz)*(b) + (blksz)*(r) + (c)])

matrix {
    int n;
    int *data;
};

matrix2 {
    int n;
    int stride;           // width of "mother" matrix
    int *data;
};

/* CELL2(matrix2,r,c) - for submatrices */
#define CELL2(m2,r,c) (((m2)->data)[((m2)->stride)*(r) + (c)])

input_params {
    int print;
    int lu2d;
    int strassen;

    int mode;
    {
        double prob[2];
        int pattern[4];
    } u;
};

void alloc_matrix(matrix *m, int n);
void naive_matrix_mult_add(matrix *C, matrix *A, matrix *B);
int parse_args(int argc, char *argv[], int *n, int *k, input_params *m_in);
void fill_matrix(matrix *m, input_params *m_in);
void print_matrix(matrix *m);
void make_matrix2(matrix *m, matrix2 *m2);
void make_submatrix(matrix2 *m, matrix2 *sub, int r, int c, int blksz);
void copy_block(int blksz, matrix *sm, int sr, int sc, matrix *dm, int dr, int dc);

void matrix_to_blocks(matrix *m, int *blockdata, int blksz, int column_first);
void blocks_to_matrix(matrix *m, int *blockdata, int blksz, int column_first);
void matrix_to_rowblocks_cyclic(matrix *m, int *blockdata, int blksz);
int intlog2(int n);
int intpow2(int power);
int count_swaps(int *reorder_all, int n);

```