```c
/*****************************************************************************
*  FILE: pmms.h
*  AUTHOR: Connor Beardsmore - 15504319
*  UNIT: OS200 Assignment S1 - 2016
*  PURPOSE: Header file for pmms.c
*  LAST MOD: 07/05/16
*  REQUIRES: stdlib.h, pthread.h, fileIO.h
*****************************************************************************/

#pragma once

#include <stdlib.h>
#include <pthread.h>
#include "fileIO.h"

//----------------------------------------------------------------------------
// CONSTANTS

#define SUBTOTAL_EMPTY 0

//----------------------------------------------------------------------------
// STRUCT: Stores the value of subtotal and the ID of the thread that
//         created it. Also stores row number that the thread calculated.

typedef struct
{
    int value;
    int rowNumber;
    long threadID;
} Subtotal;

//----------------------------------------------------------------------------
// STRUCT: Stores 3 locks for use in the producer-consumer problem.
//         Mutex provides mutual exclusion to data.
//         Full and empty are conditions that the producer and consumer
//         wait until they are met.

typedef struct
{
    pthread_mutex_t mutex;
    pthread_cond_t full;
    pthread_cond_t empty;
} Synchron;

//----------------------------------------------------------------------------
// GLOBAL VARIABLES FOR USE IN MULTITHREADS

Subtotal subtotal;
Synchron locks;
int grandTotal;
int status;

// MATRIX POINTERS AND DIMENSIONS
int* first;
int* second;
int* product;
// SEE README FOR WHAT THESE VARIABLES REPRESENT (AND REASON FOR NAMING)
int M;
int N;
int K;

//----------------------------------------------------------------------------
// FUNCTION DECLARATIONS

void* producer();
void* consumer();
int destroyLocks();
```

```
68 int createLocks();
69 void freeMatrices(int*, int*, int*);
70 void printMatrix(int*, int, int);
71 void printMatrices(int*, int*, int*, int, int, int);
72
73 //---------------------------------------------------------------------------
74
```