HW08 - Due Tue 29 March 2016 before 11:59 PM

Dr. Touma

March 24, 2016

Use HW07 as basis to breakup the source code into several files and create 2 static libraries.

1. Create 3 folders inside HW08 folder: *src*, *libs*, & *include*
2. Put the attached header files *grades_dist.h* and *stats.h* in *include*
3. Put *main.c*, *Makefile*, *grades_dist.c*, and *stats.c* in *src*. The latter two functions are implementations of the functions outlined in their respective header files.
4. Your *Makefile* should create two libraries, put them in the *libs* folder, and create the executable *grades_dis.x*. Name the libraries *libgdist.a* which depends on *grades_dist.o* and *libstats.a* which depends on *stats.o*.
5. Use the same input file *grades.csv* to test and run your code. The results should be the same as the one you obtained in HW07. In other words, run the data file on HW07, note the results and make sure HW08 gives you the same results.
6. Remove the libraries before submitting your work.
7. Your code MUST compile without errors in a) in creating the libraries, and in making the executable. Call the executable *grades_dist.x*.

```c
#ifndef STATS
#define STATS

// this is stats.h


// sorts the values of an array in ascending order
void sort_a (int *data , int size) ;

// calculates the mean of the elements of an array
float calculate_mean (const int *data, int size);

// calculates the variance of the emelemts of an array
float calculate_variance (const int *data , int size);

// calculates the median of the elemets of an array
float calculate_median (const int *data , int size );

// finds the maximum value of the elements of an array
int calculate_max ( const int *data , int size);

// finds the minimum value of the elements of an array
int calculate_min ( const int *data , int ize);


#endif
```

```c
#ifndef GRADES_DIST
#define GRADES_DIST

// this is grades_dist.h
typedef struct
{
   int min , max, data_size;
   float mean , median , std_dev ;
} Statistics ;


// gets the size of the data from file.
void get_data_size(FILE *fp, int *students, int *assignments);

// gets the data from a file.
void get_data ( FILE *fp, const int students, const int assignments, int **grades);

// bin the grades
int bin_grades (int students,  int assignments, int **grades, int **grades_scale) ;

// get stats
void get_stats (int students , int assignments, int **grades, Statistics *stats );

// print the grades
void display_grades_distribution (int assignments , int **grades_scale , Statistics *stats );

#endif
```