

## Assignment 3

Due Noon Friday October 16

The purpose of this assignment is to give you practice in writing a C program with functions and arrays. The program should be called `grader.c`

Your job is to read in an array of student grades, and for each grade determine the corresponding letter grade. Then determine the approximate median grade.

**Version 1:** Write a program that has just a main function. It should

- (a) prompt the user for number of students;
- (b) create a float array of that size;
- (c) read the values from the user; and
- (d) print out the grades one per line.

**Version 2:** Add one function:

- `char solve(float grade, int numGrades, float thresholds[], char letters[])`

This returns the letter-grade corresponding to the `grade`. The number of possible grades is contained in `numGrades`. Each threshold gives the minimum for the corresponding letter grade. Your function should assume that the array `thresholds` is **sorted in ascending order**.

For example, if called with `grade=75.6`, `numGrades=5`, `thresholds= {0,60,70,80,90}` and `letters` containing "FDCBA", the function should return 'C'.

Add to the main function some code that **before** reading the student grades: reads from the user the `numPossibleGrades`, the `thresholds` array and the letter string. Then change the code so that it now prints out each grades with the associated letter.

**Version 3:** Add at least one function:

- `float median(float list[], int count)`

This returns an approximation to the median of the list. The *median* is a value such that half the values of the list are bigger than it. You can use any approach to calculating the median, as discussed below.

Add to the main function a call to print the median at the end.

Here is sample output: Note that due to differences in implementation of the median method, your median might be slightly different.

```
Enter num grades: 5
Enter grade thresholds: 0 60 70 80 90
Enter grades (as a string): FDCBA
Enter num studs: 8
Enter grades: 60 70 80 90 100 5 75 95.5
Results are:
 60.00 D
 70.00 C
 80.00 B
 90.00 A
100.00 A
  5.00 F
 75.00 C
 95.50 A
Median is about 75.00
```

**Possibilities for calculating the median.** You can use any one of these approaches, or one of your own concoction.

1. Start with  $\text{median}=0.0$ . Repeatedly increase it by 0.1 until exactly half the values in the array are larger.
2. Sort the array. Then take the middle element.
3. Use the bisection method. The bisection method is an iterative process. At each iteration, one has an interval whose endpoints are **bottom** and **top**. One then calculates the **midpoint** of the interval. If this midpoint is too high for the desired target, then one sets **top** to **middle**, otherwise one sets **bottom** to **middle**. Every iteration shrinks the interval by half. The process terminates when the interval is sufficiently small (say less than 0.1). Then the middle of the final interval is returned.

**Submission** You are to work independently, but can ask questions from the lecturer, lab instructors and helpdesk. Late submissions will be significantly penalized.

Submit your source code using `handin.101.303 3 grader.c`