

Classwork 10: Mode

In-class Date: Monday 06 May

Due Date: Monday 13 May *May be a day or two late due to the final exam.*

Objectives

To practice working with arrays.

Assignment

Your assignment is to compute the mode of a sequence of scores stored in a file. To read from the file, you will use input redirection, which can be done as follows:

```
[arsenaul@linux1 cw10]$ ./a.out < fileToReadFrom.txt
```

Recall that the mode of a sequence of numbers, $\{x_1, x_2, \dots, x_n\}$, is the value that appears the most (ties are broken arbitrarily). For example, the mode of $\{1, 7, 2, 2, 4, 2, 1, 3\}$ is 2, since the value 2 appears the most, 3 times.

To determine the mode, you need to create a new array called `count` so that `count[i]` keeps track of the number of times that the score i has appeared. Since we will assume that the scores are between 0 and 20 (inclusive), it is good to define a constant for this value:

```
#define MAX_SCORE 20
```

To complete the program, you need to do the following:

1. Declare a new `count[]` array.
2. Initialize each element of `count[]` to 0.
3. Iterate through the elements of the `A[]` array and update `count[]` in each iteration.
 - (a) The array `A[]` holds the scores.
 - (b) For example, when your program sees that `A[34] = 11`, you should add 1 to `count[11]`, indicating another observation of that score.
4. Iterate through the `count[]` array and print out the number of times that each score appeared.
5. Iterate through the elements of the `count[]` array to find the maximum count and the score with the maximum count (the mode).
6. Print out the score that has the highest count and the number of times that score appeared.

Example Compilation and Execution

```
[arsenaul@linux1 cw10]$ gcc -Wall mode.c
[arsenaul@linux1 cw10]$ ./a.out < cw10test1.txt
count[0] is 0.
count[1] is 1.
```

```

count[2] is 2.
count[3] is 0.
count[4] is 0.
count[5] is 0.
count[6] is 0.
count[7] is 1.
count[8] is 1.
count[9] is 1.
count[10] is 0.
count[11] is 1.
count[12] is 2.
count[13] is 1.
count[14] is 0.
count[15] is 1.
count[16] is 2.
count[17] is 2.
count[18] is 1.
count[19] is 2.
count[20] is 3.
The mode of the scores is 20. It occurred 3 times.
[arsenaul@linux1 cw10]$

```

Starter Code

Use this code to help you get started. [mode.c](#)

```

/*****
** File: mode.c
** Author: <myName>
** Date: <todaysDate>
** Section: CMSC104 Section 02
** E-mail: <myEmailAddress>
**
** This file contains the main program for Classwork 10.
** The program asks the user for an array of numbers and
** tells them the average and mode.
*****/

#include <stdio.h>

#define MAX_SCORE 20
#define MAX_SIZE 1000

int main() {
    int i = 0;        // used as index into array
    int n = 0;        // number of items in array
    int r;            // used to ensure user enters an integer
    float average;    // average of numbers entered by user

```

```

float num;          // total number of items in array

int A[MAX_SIZE]; // will hold the scores

n = 0;
i = 0;

// read in the scores into array A[]
while(1) {
    r = scanf("%d", &A[i] );

    // end of input?
    if ( r <= 0 ) {
        break ;
    }

    i = i + 1;

    // exceeded array size?
    if (i >= MAX_SIZE) {
        break;
    }
}

// Make n the number of items stored in A[]
n = i;

// compute the average

int sum = 0;

for (i = 0 ; i < n ; i++) {
    sum = sum + A[i];
}

num = n; // num is float
average = sum / num;

printf("The average score is: %f\n", average);

// *****
// ***** TODO
// *****

// Declare a new count[] array.

// Initialize each element of count[] to 0. (Use a for loop.)

```

```

// Iterate through the elements of the A[] array and update count[] in
// each iteration.

// Iterate through the count[] array and print out the number of times
// that each score appeared.

// Iterate through the elements of the count[] array to find the maximum
// count and the score with the maximum count.

// Print out the score that has the highest count and the number of
// times that score appeared.

return 0;
}

```

Notes

- Remember that you need to declare constants, such as `MAX_SCORE`.
- If the largest possible score is 20, how large do you have to make the `count[]` array?
 - Hint: One larger than its highest value.
- Once you have the `count[]` array computed, how do you find the largest element in it? For example, it is not enough to know that the largest count so far is 5, you also need to remember which score has appeared 5 times.
- Two input files have been prepared for you: “cw10test1.txt” and “cw10test2.txt”. The first one is a small file with only 21 numbers that is useful for testing while you develop your program. The second file has 999 numbers.
 - `cw10test1.txt`
 - `cw10test2.txt`

Extra Credit

Count how many invalid scores there are in the input file. Do this by adding a check for invalid scores in the loop you wrote for Step 3. Then, after printing the mode, print how many invalid scores there were, if any.

Grading Rubric

- Header comment: 2 points
- Body comments: 3 points
- Compiles: 40 points
- Uses redirection to test: 10 points
- Displays correct Count array: 25 points

- Finds mode: 15 points
- Lists how often mode occurred: 5 points
- EC: +5 points

What to Submit

Use the `script` command to record yourself compiling and running your programs 3 times, using different numbers each time. (Do not record yourself editing your program!) Exit from `script`. Submit your programs and the typescript file.

```
[arsenaul@linux1 cw10]$ submit cmsc104_arsenaul cw10 mode.c typescript
```

Verify

Make sure you submitted the assignment correctly.

```
[arsenaul@linux1 cw10]$ submitls cmsc104_arsenaul cw10
```