

**Objectives:**

Use strings, files, and command line arguments.

You are asked to grade a multiple-choice test for a class. The students' answers are stored in a file. Each student record in the file includes a student's first name, last name, and the test answers given by the student. A blank space indicates that the student did not answer the question. For example, the following line represents a student's record.

George Bush cCDdaab ccaADdc



In the above sample input, George Bush did not answer question 8.

The test answer key is entered in the first line of the input file (ACDBCBDCACDACC).

Write a C++ program that determines the scores and letter grade of students' multiple-choice tests. The number of test questions must be determined by reading the answer key string at the beginning of the input file. The number of letters in the answer key determines the number of questions on the exam. In other words, the number of questions is not constant. Generate the report shown in the sample output and store it in an output file.

**Output the following:**

- A grade report for each student **exactly** as it appears in the sample output below. Replace a missing answer with a '-'.
- The test average and letter grade.
- **(Extra Credit):** The names and averages of all the students with the highest grades. No hints will be given for the extra credit.

Your program must use command line arguments (see below) to get the input and the output file names. Exit the program with an error message if any of the files fails to open or the user enter the wrong number of arguments. The program can be run using the following command:

```
./a.out exam.txt report.txt
```

If your file name contains spaces surround the file name with double quotes as in:

```
./a.out "input file name.txt" "output file name.txt"
```

The following should produce an error and exit the program:

```
./a.out exam.txt  
./a.out exam.txt report.txt report2.txt  
./a.out
```

**Required Functions:**

- Write a function that grades a single student score and returns the score.
- Write a function that returns a letter grade for a score.
- Write a function that outputs a single student's report to the screen.

**Parsing command line arguments (Section 6.6 in your textbook)**

In C++ you can input data (strings) into your program on the command line (command line arguments). You can capture these data by adding two parameters to your main program as follows:

```
int main (int argc, char *argv[])
```

`argc` (argument count): stores the number of arguments on the line, including the name of the program

`argv` (argument vector): Is an array that stores all the strings entered at the command line including the program name.

For example, executing the command:

```
./a.out exam.txt report.txt
```

Assigns:

```
argc = 3
```

```
argv[0] will be assigned the string "./a.out"
```

```
argv[1] will be assigned the string "exam.txt"
```

```
argv[2] will be assigned the string "report.txt"
```

To add the above arguments to your program, just assign them to a string object. For example:

```
string inputFileName;  
inputFileName = argv[1];  
etc.
```

**Hints:**

- Start by reading the input file and see if you can display its content (key, first, last, answers).
- Use command line arguments to achieve the previous step.
- Check for command line arguments errors.
- Write each function and test it before moving the next question.
- Use the `at()` member function of the string to access each test question answer or each answer key.
- You can find the number of characters in a string using the member function `length` (example: `someString.length()`)
- The function `getline()` reads a string that include white spaces.

**Sample input file (Provided in the \_TEST directory)**

```
ACDBCBDCCACDACC
George Washington aDdbcBdc accAcc
John Adams   cdb bd cac ac
Thomas Jefferson abcd daacbbcd dda
Ronald Regan bcd bcbDccACdacc
John Kennedy cbbadcb bacdbadc
Jimmy Carter abdcad abdc bacd
George Bush  CCDDAABBCCAADD C
Barack Obama acdbcb dcccac dacc
Bill Clinton ACDBCBDCCACAACC
Richard Nixon acdbcacccac dacc
```

**Sample output file report (Note: the averages are just samples and may not be accurate for the given data):**

Washington, George

-----  
Answers, correct answer in parenthesis

```
1: a(a)   2: d(c)   3: d(d)   4: b(b)   5: c(c)
6: b(b)   7: d(d)   8: c(c)   9: -(c)  10: a(a)
11: c(c)  12: c(d)  13: a(a)  14: c(c)  15: c(c)
```

Score: 80.0% B

-----  
Adams, John

-----  
Answers, correct answer in parenthesis

```
1: -(a)   2: c(c)   3: d(d)   4: b(b)   5: -(c)
6: b(b)   7: d(d)   8: -(c)   9: c(c)  10: a(a)
11: c(c)  12: -(d)  13: a(a)  14: c(c)  15: -(c)
```

Score: 66.7% D

-----  
...  
...  
-----

Class average: 76.4% C

-----  
Students with the highest grade (100%):

Barack Obama  
Richard Nixon

**Grading:**

Programs that contain syntax errors will earn zero points.

Programs that do not include functions, other than the main, will earn zero points.

Programs that use any library that was not discussed in class will earn zero points.

**(45 points)**

- (7 points) for each of the three required functions.
- (19 points)
  - Correctness
  - User interface
  - Clarity and format of the output

**(5 points)**

- Programming Style & documentation.

**Extra credit: (5 points)**

Follow the coding style outline on GitHub:

- <https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md>