CS 101 **Project 1** **Summer 2016**

Write a C++ program project1.cpp that operates as described below.

Your program should accept three command line arguments. Argument 1 is the number of rows, and argument 2 is the number of columns. Argument 3 is the name of an input file, and argument 4 is the name of an output file.

Display your first name, your middle name, and your last name, each in a rectangular formation as indicated by the specified numbers of rows and columns. Also, display an increasing number of dots before, between, and after repetitions of the name on the same row. Next, repeat the same formation for each person's name in the input file. Finally, repeat the same formation for your name again.

Each program should behave as shown in the following example using 2 rows and 5 columns, which was submitted by an imaginary student Maria Jeanne Davis.

Input file

```
George Washington Carver
Mary Tyler Moore
Edgar Allan Poe
```

Output file

```
.Maria..Maria...Maria....Maria.....Maria......
.Maria..Maria...Maria....Maria.....Maria......

.Jeanne..Jeanne...Jeanne....Jeanne.....Jeanne......
.Jeanne..Jeanne...Jeanne....Jeanne.....Jeanne......

.Davis..Davis...Davis....Davis.....Davis......
.Davis..Davis...Davis....Davis.....Davis......


.George..George...George....George.....George......
.George..George...George....George.....George......

.Washington..Washington...Washington....Washington.....Washington......
.Washington..Washington...Washington....Washington.....Washington......

.Carver..Carver...Carver....Carver.....Carver......
.Carver..Carver...Carver....Carver.....Carver......


.Mary..Mary...Mary....Mary.....Mary......
.Mary..Mary...Mary....Mary.....Mary......
```

```
.Tyler..Tyler...Tyler....Tyler.....Tyler......
.Tyler..Tyler...Tyler....Tyler.....Tyler......

.Moore..Moore...Moore....Moore.....Moore......
.Moore..Moore...Moore....Moore.....Moore......


.Edgar..Edgar...Edgar....Edgar.....Edgar......
.Edgar..Edgar...Edgar....Edgar.....Edgar......

.Allan..Allan...Allan....Allan.....Allan......
.Allan..Allan...Allan....Allan.....Allan......

.Poe..Poe...Poe....Poe.....Poe......
.Poe..Poe...Poe....Poe.....Poe......


.Maria..Maria...Maria....Maria.....Maria......
.Maria..Maria...Maria....Maria.....Maria......

.Jeanne..Jeanne...Jeanne....Jeanne.....Jeanne......
.Jeanne..Jeanne...Jeanne....Jeanne.....Jeanne......

.Davis..Davis...Davis....Davis.....Davis......
.Davis..Davis...Davis....Davis.....Davis......
```

When displaying the first names, you must use three nested for-loops. When displaying the middle names, you must use three nested while-loops. When displaying the last names, you must use three recursive functions; that is, use recursion to replace iteration. Programs that do not follow these requirements will not receive credit.

If you don't have a middle name, then you may invent any middle name you choose.

Please carefully read the following requirements:

- You must do your own work; you must not share any code. If you violate this rule, you may receive an invitation to the dean's office to discuss the penalties for academic misconduct.
- Make sure your program runs properly on cs-intro.ua.edu. Your program will be graded on that system.
- Submit your project in a zipfile that contains only your project1.cpp file. There should be no subdirectories or extra files.
- If you violate the requirements such that it breaks our grading script, your project will be assessed a significant point deduction, and extreme or multiple violations may cause the project to be considered ungradable.

- Two pairs of example input/output files and a test script are provided. Run the following commands to test your program. If any differences appear (other than using your name instead of imaginary student Maria Jeanne Davis), then your project does not match the specifications. Points will be deducted for each incorrect line of output.

  ```
  unzip *.zip
  chmod u+x testscript.sh
  ./testscript.sh
  ```

- Alternatively, instead of running the test script, you can instead enter these commands.

  ```
  g++ project1.cpp –Wall –lm –o project1
  ./project1 2 5 input0.txt temp0.txt
  diff output0.txt temp0.txt
  ./project1 3 6 input1.txt temp1.txt
  diff output1.txt temp1.txt
  ```

- Submit your project on Blackboard by the due date (11:59pm Friday). There is a grace period of 24 hours (until 11:59pm Saturday). Projects submitted on Sunday will be assessed a late penalty of 5% per hour. No projects will be accepted after Sunday.
- Double-check your submission when you submit it. Errors discovered later cannot be fixed and resubmitted after the project is graded. Projects will not be re-graded unless an error is found in the grading script or in the input/output files used during grading.