

San José State University  
Department of Computer Engineering

CMPE 180-92  
**Data Structures and Algorithms in C++**  
Fall 2016  
Instructor: Ron Mak

**Assignment #3C**

**Assigned:** Sunday, September 11  
**Due:** Friday, September 16 at 11:59 PM  
**URL:** <http://codecheck.it/codecheck/files/1609110713np55ladz8dqtmx5la97l669o>  
**Canvas:** Assignment 3.c. War and Peace  
**Points:** 100

**War and Peace**

This assignment will give you practice with C++ strings and vectors.

Download and unzip the complete text of *War and Peace* as an ASCII file from  
<http://www.cs.sjsu.edu/~mak/CMPE180-92/assignments/3C/WarAndPeace.zip>  
The file contains over 65,000 lines and over a half million words.

Write a C++ program to search for the following names in the text:

- Makar Alexeevich
- Joseph Bazdeev
- Boris Drubetskoy

For each occurrence of each name, print:

- The starting line number (first line is 1).
- The character position of the first letter of the first name (first position is 1).
- The name itself.

You must print the names in the order that they appear in the file. If a line contains multiple names, print them in alphabetical order by last names.

A name can be split across two consecutive lines. More than one name can be in a line. You may assume that the names are always in the order of the first name followed by the last name. Between the first and last names, there is either a single space or a line break.

## Strings and vectors

You must use C++ strings and not the old C strings, and you must use vectors instead of arrays. Use the vector `at` method instead of indexing with `[ ]`.

For string methods, see <http://www.cplusplus.com/reference/string/string/>

For vector methods, see <http://www.cplusplus.com/reference/vector/vector/>

## Expected output

LINE	POSITION	NAME
19949	1	Boris Drubetskoy
21906	66	Makar Alexeevich
21953	2	Makar Alexeevich
22173	9	Boris Drubetskoy
23910	62	Boris Drubetskoy
23982	39	Boris Drubetskoy
24033	54	Boris Drubetskoy
29439	31	Boris Drubetskoy
29614	28	Boris Drubetskoy
33007	1	Boris Drubetskoy
41110	22	Boris Drubetskoy
46593	60	Joseph Bazdeev
46612	19	Joseph Bazdeev
46622	28	Makar Alexeevich
46626	18	Makar Alexeevich
46626	39	Joseph Bazdeev
46642	20	Joseph Bazdeev
46689	11	Makar Alexeevich
48300	11	Makar Alexeevich
48314	42	Makar Alexeevich
48317	10	Makar Alexeevich
48323	68	Makar Alexeevich
48326	38	Makar Alexeevich
48331	60	Makar Alexeevich
48336	13	Makar Alexeevich
48347	28	Makar Alexeevich
48407	53	Makar Alexeevich
48409	19	Makar Alexeevich
48415	8	Makar Alexeevich
48433	24	Makar Alexeevich
48435	1	Makar Alexeevich
48468	68	Makar Alexeevich

## Tips

Write the first version of your program to find only names contained in a single line. Your second version can find names that are split across two lines.

If you search for the names in alphabetical order and there is more than name in a line, you will find them in alphabetical order.

## Rubrics

Criteria	Maximum points
<b>Good output</b> (as determined by CodeCheck) <ul style="list-style-type: none"><li>• Correct output values.</li><li>• Correct output formatting.</li></ul>	<b>30</b> <ul style="list-style-type: none"><li>• Values: 20</li><li>• Formatting: 10</li></ul>
<b>Good program design</b> <ul style="list-style-type: none"><li>• Good use of C++ strings</li><li>• Good use of vectors.</li><li>• Good use of functions.</li><li>• Good choice of names.</li></ul>	<b>50</b> <ul style="list-style-type: none"><li>• Strings: 15</li><li>• Vectors: 15</li><li>• Functions: 10</li><li>• Names: 10</li></ul>
<b>Good program style</b> <ul style="list-style-type: none"><li>• Follow the formatting (indentation, position of braces, etc.) and naming conventions of the Savitch textbook.</li><li>• Greatly reduce, or eliminate entirely, the use of global variables. Use function parameters instead. Global constants are OK.</li></ul>	<b>20</b> <ul style="list-style-type: none"><li>• Formatting: 10</li><li>• Global variables: 10</li></ul>

You can submit as many times as necessary to get satisfactory results, and the number of submissions will not affect your score. When you're done with your program, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution.

Submit the signed zip file into **Canvas: Assignment 3.c. War and Peace**.