

CAB201 Programming Principles - Semester 2, 2019

Report for Assignment: Project – Genomic Sequence Retrieval - Part II

Student name and number:

[James Norman n10256989]

Build and Run Instructions

Please provide clear step-by-step instructions here on how to build your program in Visual Studio and run your program in the command line, given your submitted zip folder. For each step, you should include a screenshot. You may expand the box if needed.

[Enter build & run instructions. Include where to find .sln file (with screenshot), how to build your program (with screenshots), how to find the .exe file (with screenshots), and finally a screenshot of your .exe program being run from the command line]

Make sure that these instructions are tested as follows:

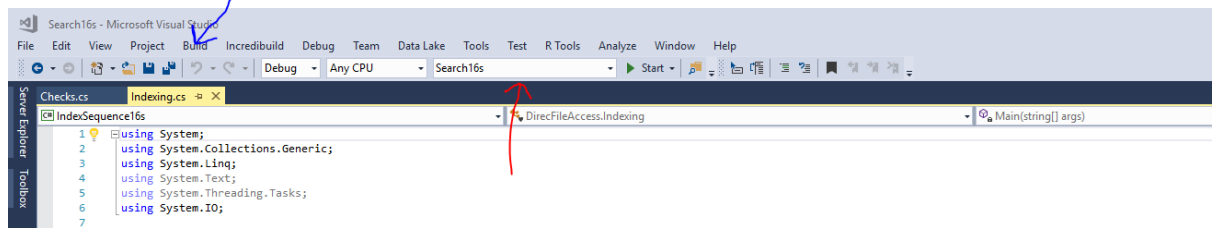
1. Place your project (exactly what you are about to upload as a submission) on a USB disk
2. Log onto a QUT SEF computer lab – this must be a **new login session**
3. Follow your own build & run instruction
4. Test your program.

Note that this is what the markers will do – if your program fails to build and execute you will lose all marks for testing your code.

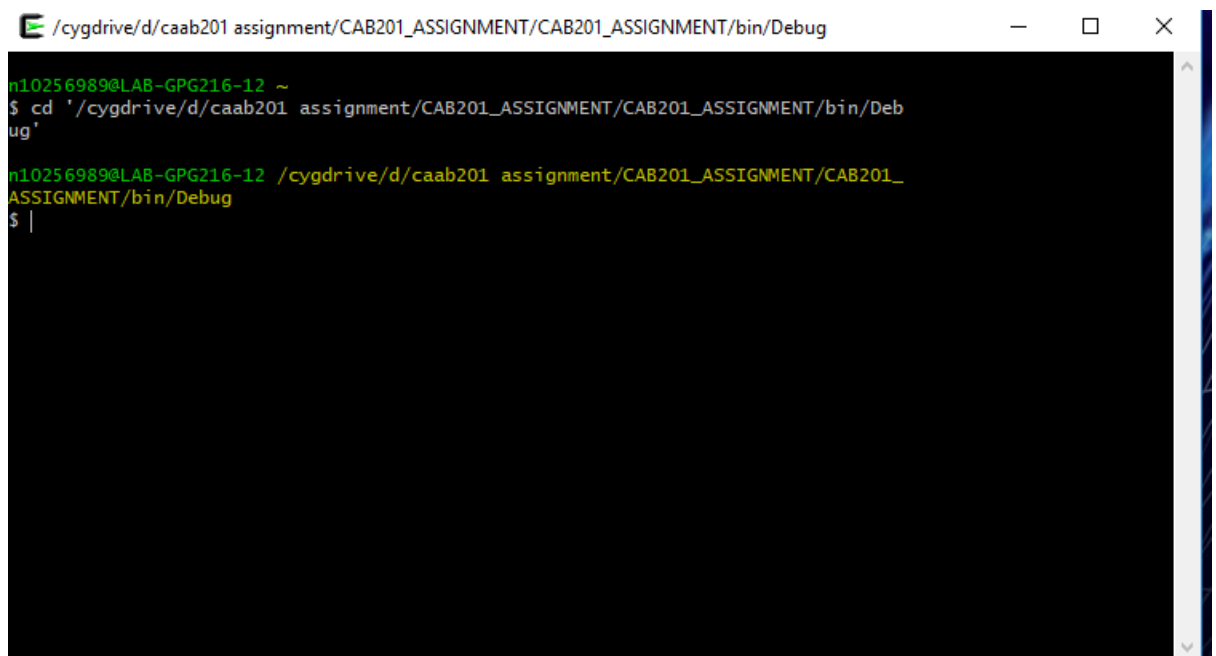
IF THE PROJECT DOES NOT BUILD & RUN YOU MUST DECLARE IT HERE

AS WELL AS IN THE STATEMENT OF COMPLETENESS

1. Unzip Cab201Assignment folder
2. Open Search16s.sln
3. Build both the indexing and main search 16s program (build is located at the blue line and the way to change build is in red)



4. Open command prompt and navigate to the debug section of the folder where the .exe is



5. Run the program as you wish using the command prompt

```

/cygdrive/d/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug
10256989@LAB-GPG216-12 ~
$ cd '/cygdrive/d/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug'
10256989@LAB-GPG216-12 /cygdrive/d/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug
$ ls
16S.fasta  16S.index  CAB201_ASSIGNMENT.exe.config  query.txt
16S.index  CAB201_ASSIGNMENT.exe  CAB201_ASSIGNMENT.pdb
10256989@LAB-GPG216-12 /cygdrive/d/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug
$ ./CAB201_ASSIGNMENT.exe IndexSequence16s 16S.fasta 16S.index

```

Statement of Completeness

This statement of completeness will need to *accurately* state the functionality which has been implemented. There will be a penalty of 3.5 marks (loss of 3.5 marks) for a non-completed or submitted statement of completeness, and a penalty of 1 mark for each inaccurate statement to a maximum of 3 marks.

In the following section, you are required to mark which functionality you have implemented. In the column on the right please mark 'Y' where you have completed this functionality, and 'N' where you have not. Please fill in any additional text boxes requested, and please note any limitations or bugs in the box at the end of each section. You may expand the table if you need more room for comments.

Basic Functionality		
Build & Run	When following the Build & Run instructions, the program successfully builds and runs. This was tested in a QUT SEF lab with a new login session, using the same zipped folder that is submitted.	Y
Basic itinerary output	The program displays the data from the file	Y
	The program displays the appropriate line	Y
	The correct amount of information is displayed, e.g. only the relevant entries	Y
	The correct level, provided as a command line flag - <i>levelN</i> , is executed	Y
	The program does not store the whole file in memory, instead it accesses the file on disk	Y
Error handling	A clear error message is displayed when an incorrect number of arguments is provided	Y

	A clear error message is provided when an incorrect flag is provided (e.g. not -level1, etc.)	Y
	A clear error message is provided when the input file doesn't exist, or is incorrectly formatted	Y
Comments		

Searching Algorithm – Part II

Please underline, circle or highlight the levels that were completed.

Algorithm	Level: Level 4, level 5, level 6, level 7
	Bonus: Sequence matching using wildcards <i>If completed, please highlight level 7 above and discuss how you approached the problem and why you solved it the way you did.</i>
	Level 7 Explanation <i>I used .replace and regular expression (REGEX) to replace characters from the sequence and then see if they match to produce the desired output. I mostly just used google as my resource.</i>
Comments	My levels use .contain so it brings up all sequences that contain the input and not just ones that completely match.

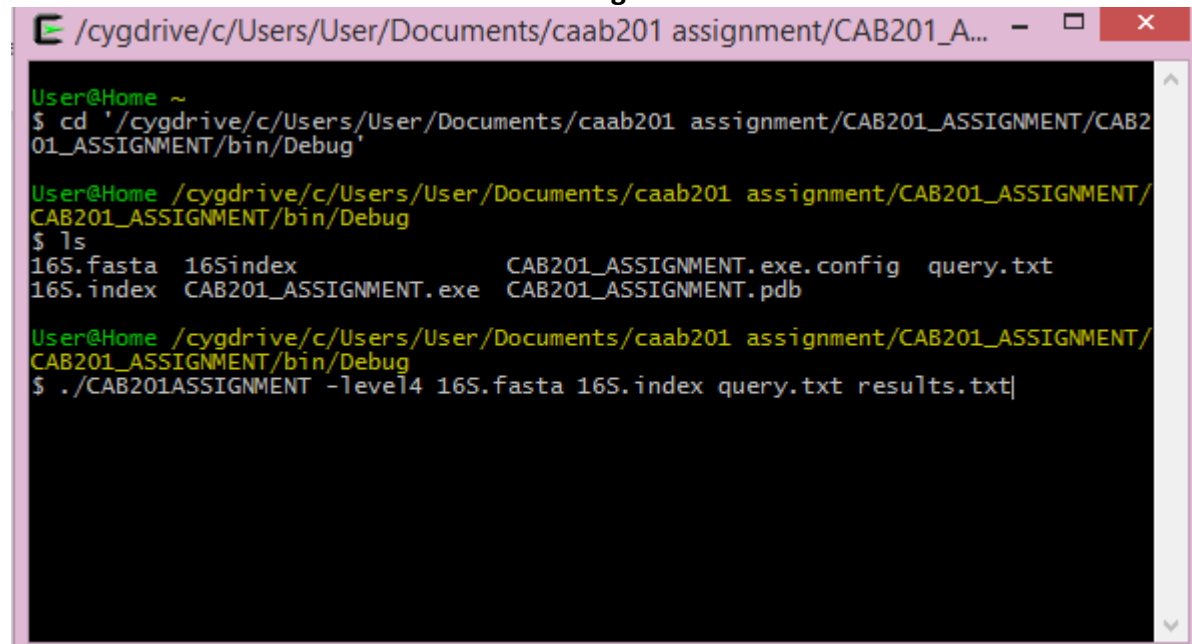
Screenshots of Functionality

In the following section, you are required to provide screenshots that provide evidence of your program working with provided input. **You must complete this section.**

- 1) The 16S.fasta file has been provided with this template. Download them and place them in the same folder as your .exe file. You may have extra files, e.g. a query file, in this folder, and your .exe may be named differently. This is fine.
- 2) Open the command prompt and go to the above folder. In the command line, type the name of the .exe file and copy and paste following arguments:

Search16s -level4 16S.fasta 16S.index query.txt results.txt

Place screenshot of entered command line arguments here:



```
/cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_A... - □ ×  
User@Home ~  
$ cd '/cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug'  
User@Home /cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug  
$ ls  
16S.fasta 16S.index CAB201_ASSIGNMENT.exe.config query.txt  
16S.index CAB201_ASSIGNMENT.exe CAB201_ASSIGNMENT.pdb  
User@Home /cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_ASSIGNMENT/CAB201_ASSIGNMENT/bin/Debug  
$ ./CAB201ASSIGNMENT -level4 16S.fasta 16S.index query.txt results.txt|
```

- 3) Hit enter to run your program.

Place screenshot(s) of the full output to console of your program. You may expand the box as necessary, and use as many screenshots as needed:

```
/cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_A... - □ ×  
User@Home /cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_ASSIGNMENT/  
CAB201_ASSIGNMENT/bin/Debug  
$ ./CAB201_ASSIGNMENT -level4 16S.fasta 16S.index query.txt results.txt  
Error sequence NR_999999.9 not found  
User@Home /cygdrive/c/Users/User/Documents/caab201 assignment/CAB201_ASSIGNMENT/  
CAB201_ASSIGNMENT/bin/Debug  
$ |
```


Self-Assessment:

1) How do I think I went with this assignment?

I think I did fairly well in this assignment getting levels 4-6 done in a timely fashion, the main problem I will have with marking will be nit picking where I'll probably lose 1-2 marks.

2) What did I find difficult in this assignment?

Level 4 proved very difficult in terms of understanding the indexing file after we'd been doing it with the normal 16S

3) What would I do differently next time?

Next time I would make sure to ask my tutor about everything because I assume the tutors are going to be pedantic about things like leaving > in the id number.

4) Were there any bugs in my assignment, if so what were they?

The only bug I detected in my program is that it brings up all the id's that contain a certain sequence/name instead of completely matching it.

CRA:

Please fill out the following CRA, reporting how many marks you believe your project might be awarded. Your assessment should be a considered reflection on what you have achieved. The purpose of this is to advise the marker of what you believe was achieved in order for us to pay attention to discrepancies. **Your self-assessment is NOT attracting marks, but must be provided (penalty applies if missing).**

<u>Code Quality</u>	
To score points in this section, the student must follow the code quality guidelines as specified in the C# Coding Style Guide on Blackboard	29/30
Maintained consistent, clear, and meaningful standard in variable and method naming. No magic numbers.	3/3
Well structured – consistent and appropriate white spacing, line length, indentation, and separation into files within the project (i.e. one class per file)	2/2
Well commented – class header comment at beginning of each class, comment before every method, and in-line comments to explain complex or not easily discernible code. In-line comments are not excessive.	4/4
The DRY principle (Don't repeat yourself) is followed where appropriate	2/3
Methods are single purpose and clear	4/4
Classes are well designed, with high cohesion and low coupling	8/8
Classes are separated into reusable modules where appropriate	3/3
Exceptions are thrown and handled appropriately	3/3

<u>Basic Functionality</u>	
To score marks in this section, your program must be able to be run from the command line with the appropriate arguments.	14/15

<i>Basic Output</i>	The program displays the data from the file	1/1
	The program displays the appropriate line	1/1
	The correct amount of information is displayed, e.g. only the relevant entries	1/1
	The correct level, provided as a command line flag <i>-levelN</i> , is executed	1/1
	The program does not store the whole file in memory, instead it accesses the file on disk	5/5
	Total:	9/9
<i>Error Handling</i>	A clear error message is displayed when an incorrect number of arguments is provided	2/2
	A clear error message is provided when an incorrect flag is provided (e.g. not -level1, etc.)	2/2
	A clear error message is provided when the input file doesn't exist, or is incorrectly formatted	1/2
	Total:	5/6

<u>Part II</u>		Marks Available:
To score marks in this section, your program must be able to run levels 4-7.		55/55
<i>Level 4</i>	The program creates a file as specified by the command line arguments	2/2
	The index file contains a list of all the sequence ids with the appropriate byte-offset	5/5
	The searching program makes use of the created index file to execute a number of queries	5/5
	A clear error message is provided when the index file does not exist	1/1

	Clear error messages are provided when the query file cannot be found, or when a bad query is given, like in Level 3	2/2
	Total:	15/15
<i>Level 5</i>	The program correctly locates and prints the requested sequence ids	10/10
	A clear error message is provided when the sequence does not exist	5/5
	Total:	15/15
<i>Level 6</i>	The program correctly locates and prints the requested sequence ids	15/15
	A clear error message is provided when the keyword does not exist	5/5
	Total:	20/20
<i>Level 7 (Optional)</i>	The program correctly decodes the expression given and identifies any matching sequences	5/5
	A clear error message is provided when the sequence does not exist	5/5
	Total:	10/+10

Test planning

Category	Input	Expected result	Actual output	Passed
Wrong amount of inputs	-level4 16S.fasta	Please enter more inputs	Please enter more inputs	Y
Wrong files	-level4 315 1	Please enter a valid file name	Please enter a valid file name	Y
Wrong files	4 16s.index	Please enter a valid file name	Please enter a valid file name	Y
Wrong amount of inputs	16S.fasta	Please enter more inputs	Please enter more inputs	Y
Wrong amount of inputs	-level 4 16S.fasta Query.txt 16S.index results.txt p	Please enter less inputs	Please enter less inputs	Y
Index file not found	-level4 16S.fasta Query.txt 16S.index results.txt	Index file not found	Index file not found	Y
Wrong files	-level 5 16S.fast	Please enter a valid file name	Please enter a valid file name	Y
Level 4 working output	-level4 16S.fasta Query.txt 16S.index results.txt	Sequence >NR_9999999 Was not found	Sequence >NR_9999999 Was not found	Y
Level 5 found	Search16s -level5 16S.fasta CTGGTACGGTCAACTTGCTCTAAG	NR_123456.1 NR_118941.1 NR_432567.1	NR_123456.1 NR_118941.1 NR_432567.1	Y
Level 6 found	Search16s -level6 16S.fasta Streptomyces	NR_026530.1 NR_026529.1...	NR_026530.1 NR_026529.1....	Y
Level 5 not found	Search16s -level5 16S.fasta pop	Sequence pop Was not found	Sequence pop Was not found	Y
Level 4 found	-level4 16S.fasta Query.txt 16S.index results.txt	Please press enter to continue	Please press enter to continue	Y
Level 6 not found	Search16s -level6 16S.fasta pop	Sequence pop Was not found	Sequence pop Was not found	Y