GEDT019 Final Assignment

Instruction

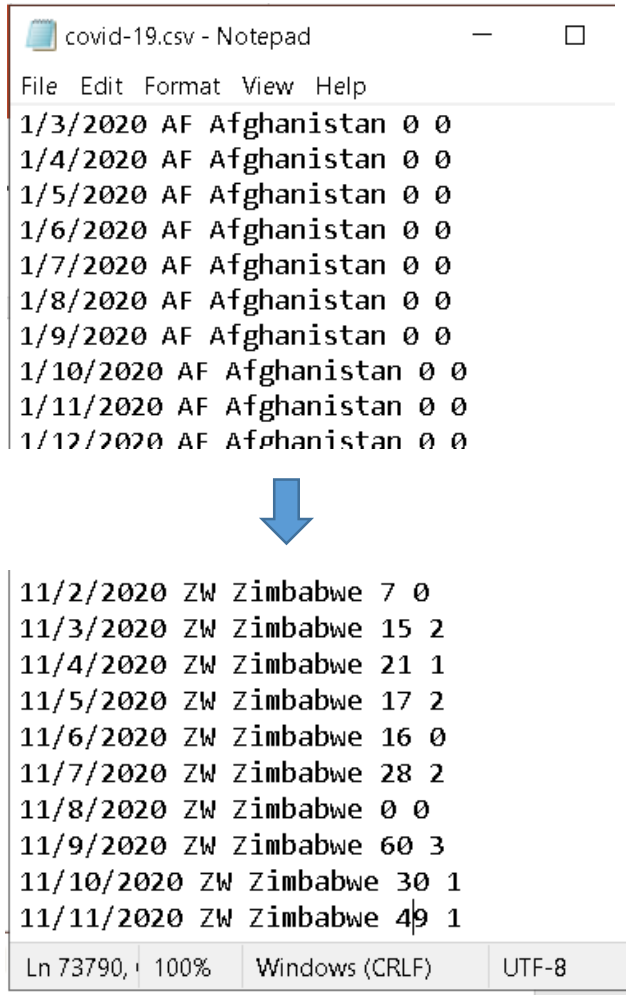
- **Read this instruction very carefully.**
- You will name the filename as *[StudentIDNumber]*.cpp or .cc.
  - Submit the file *[StudentIDNumber]*.cpp or .cc on iCampus before the deadline.
- You should submit the source code only.
- You may submit partial solution (and you'll receive partial credit).
- You may ask questions thru email at atang@skku.edu.
  - But please ask your questions early (otherwise, I may not reply in time before your deadline)
- I will make clarifications about the problem on icampus thru QnA.
- You may submit multiple versions.
  - I will grade the last version before the deadline you submitted only.
  - Work submitted after the deadline will not be graded
- Make backup!
- The deadline is Wednesday 5th December 2021 23:59 pm.

Collaboration Policy

- The work you submit must be the work of your own.
- You are free to give or receive help when doing homework assignments, but you must follow the following restrictions:
- Only the helper can look at the code of others. Student who is receiving help must not look at the code of the helper;
- Student who is receiving help must do all the typing herself/himself. Helper must not touch the computer of the student who is receiving help; and
- All student can not post your code on the web, nor send your code to other students.

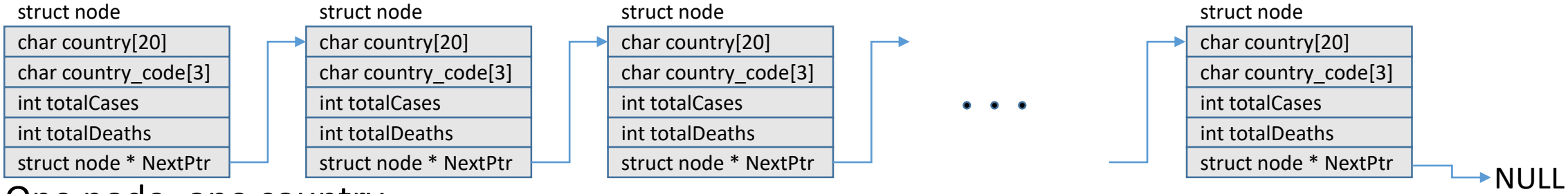
Introduction

- In this assignment, you will write a program to process a data file according to some specification.
- You are provided with a sample data file named "`covid-19.csv`".
- This file consist of covid-19 data downloaded from WHO:
  - This file is in csv format (with a space " " as common separator)
  - The file contains data from
    - Column 1 – Date
    - Column 2 – Country code
    - Column 3 – Country
    - Column 4 – New infection cases on that date
    - Column 5 – New death cases on that date
    - e.g. on 3 Jan 2020, there is 0 new case and 0 new death in Afghanistan



The Task

- You are supplied with a code.
- You need to build your code based on the supplied code.
- What you need to do:
  - Read data from an input file (“covid-19.csv”)
  - Calculate total infection cases and total death cases for each country
  - Form a linked list as follow:



  - One node, one country
  - totalCases represents the total infection cases from day 1 to the last day in the data for one country
  - totalDeaths represents the total death cases from day 1 to the last day in the data for one country
  - NextPtr points to the node for next country
  - NextPtr points to NULL if it is the last node

The Task (cont.)

- After you constructed the linked list, you need to:
  - Search the linked list for the country with maximum infection case, and then point the pointer "`node * maxTotalCases`" to this node
  - Search the linked list for the country with maximum infection case, and then point the pointer "`node * maxTotalDeaths`" to this node
  - Implement the 5 functions:
    - `int Length(node * chain)`
      - This function takes a pointer to a linked list as parameter, and calculate the length (i.e. number of node) in the linked list as return value.
    - `void PrintALL(node * chain)`
      - This function takes a pointer to a linked list as parameter, and then print out all the data in the linked list. This function does not have any return value.
    - `node * FindCountryCode(node * chain, char * input)`
      - This function takes a pointer to a linked list and a character string as input, and then find if there is any node that matches the country_code with the character string. The function returns a pointer pointing to the node if a match is found, and returns 0 if not found.
    - `node * FindMaxCases(node * chain)`
      - This function takes a pointer to a linked list and a character string as input, and then find node with the maximum Number of Cases. The function returns a pointer pointing to the node with the max number of cases.
    - `node * FindMaxDeaths(node * chain)`
      - This function takes a pointer to a linked list and a character string as input, and then find node with the maximum Number of Deaths. The function returns a pointer pointing to the node with the max number of deaths.
  - **You should not use any Global Variable in your code.**
  - **You should not modify the existing code.**
  - **Please avoid using external libraries unless it is necessary**
    - **It's okay to use libraries that we covered in the course**

Assumptions

- Regarding the datafiles:
  - The data file provided is only a sample data file. Your program should work with any data file with the same format
  - You should not assume that there is 235 countries in the data files (i.e. I may test your code with data file containing 3 countries, or 300 countries)
  - You may assume that the maximum number of countries is 500, and the minimum number of country is 1.
  - You should not assume that the file contains 314 days of data (i.e. I may test your code with data file containing 3 days of data, or 500 days of data)
  - You may assume that data for each country has the same number of day
    - i.e. If there are 30 days of data for country A, then there are 30 days of data for all countries in the data file
  - You may assume that data for each country are sorted according to the date in ascending order
    - i.e. if there are 314 days of data in the file, Line 1-314 contain the data for the first country listed in ascending order according to the date, Line 315-628 contain the data for the second country listed in ascending order according to the date, etc).
  - You are supplied with 3 sample data files for testing purposes.
  - You may open the data file and calculate the results in Excel, so that you can compare the results with the results generated by your program

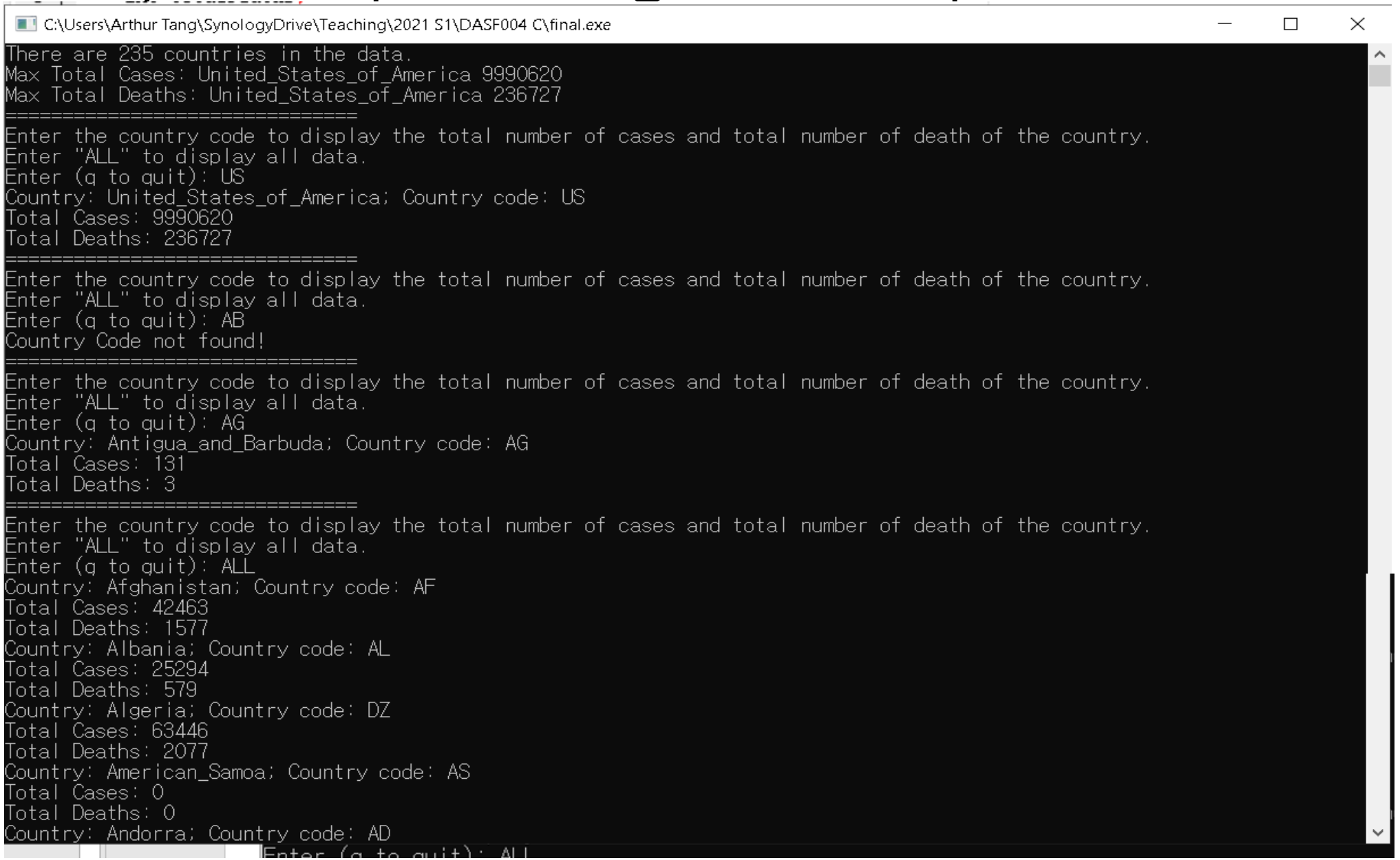
final.cpp [*] finalA.cpp

```
11
12  int Length(node * chain)
13  {
14  }
15
16  void PrintALL(node * chain)
17  {
18  }
19
20  node * FindCountryCode(node * chain, char * input)
21  {
22  }
23
24  node * FindMaxCases(node * chain)
25  {
26  }
27
28  node * FindMaxDeaths(node * chain)
29  {
30  }
31
32  int main(void)
33  {   FILE * fPtr = fopen("covid-19.csv","r");
34      char currentCountry[80] = "";
35      int counter=0;
36      node chain[235];
37      node * maxTotalCases=0;
38      node * maxTotalDeaths=0;
39      node * head=&chain[0];
40
41      /* Fill in your code here.
42      Your code should:
43          - Read data from the data file ("covid-19.csv").
44          - Calculate the total infection cases and total death cases for each country.
45          - Construct a linked list based on the calculated results.
46      */
47
```

Make these 5 functions

Fill in your code here

Sample output using the sample data file



C:\Users\Arthur Tang\SynologyDrive\Teaching\2021 S1\DASF004 C\final.exe

```
Total Deaths: 830
Country: Viet_Nam; Country code: VN
Total Cases: 1226
Total Deaths: 35
Country: Wallis_and_Futuna; Country code: WF
Total Cases: 1
Total Deaths: 0
Country: Yemen; Country code: YE
Total Cases: 2070
Total Deaths: 602
Country: Zambia; Country code: ZM
Total Cases: 16997
Total Deaths: 350
Country: Zimbabwe; Country code: ZW
Total Cases: 8610
Total Deaths: 255
==============================
Enter the country code to display the total number of cases and total number of death of the country.
Enter "ALL" to display all data.
Enter (q to quit): ABC
Country Code not found!
==============================
Enter the country code to display the total number of cases and total number of death of the country.
Enter "ALL" to display all data.
Enter (q to quit): AQ
Country Code not found!
==============================
Enter the country code to display the total number of cases and total number of death of the country.
Enter "ALL" to display all data.
Enter (q to quit): q
```

- A set of testing input files was provided to you for testing purpose, but your program should work with any input files according to the specifications
- Submit your source code to iCampus before Wednesday 5 December 2021 23:59 pm
- You may submit multiple version, and only the latest version will be graded
- Late submission will not be graded and a zero score will be given
- In case of technical difficulties for iCampus server, you may submit it as email attachment before the deadline (atang@skku.edu)
- You should make back up to your work, just in case of accident.
- You should double check your submission. Wrong submission will not be an excuse for late submission.