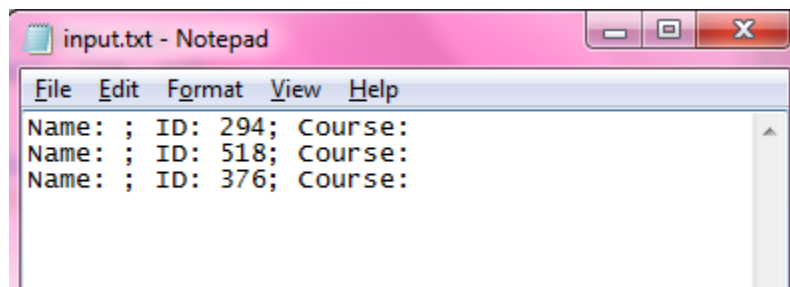


Lab 7

In Lab 7, you are required to write a program that input/output data from data files and manipulate strings. It contains **2** parts, which are demonstrated in below contents.

Part I (3.5 points)

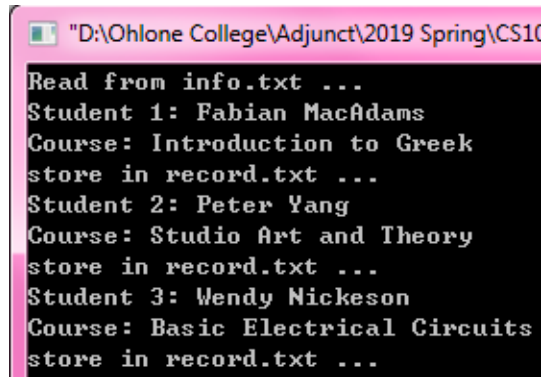
This part will read data from *info.txt*, which contains 3 lines of students' information including student name, student ID and course name.



Figure_1 Data in input.txt

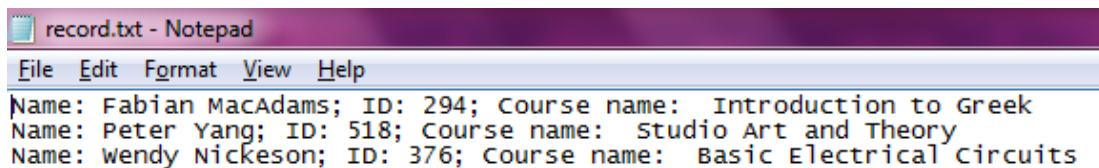
Follow below steps to fill in students' information and store updated information into *record.txt*.

1. Download and save *info.txt* under the same directory as the source file does;
2. Open the file in *input* mode and check the open status; if open successfully, read the line one by one until it fetches all characters;
3. Setup a counter to count number of lines/students; prompt user for the number of student (e.g., student1, student2, student3);
4. Call *replace* function to replace "Course: " with "Course name: " (notice there is a space after semicolon);
5. Create a string and store user input student's name one by one; call *insert* function to insert each student's name into the correct position;
6. Create another string and store user input course name one by one; concatenate current student information with each course name one by one;
7. Create and open *record.txt* in *output* mode, write the modified student's information into this file line by line;
8. Close all files when finishing above operations.



```
"D:\Ohlone College\Adjunct\2019 Spring\CS10
Read from info.txt ...
Student 1: Fabian MacAdams
Course: Introduction to Greek
store in record.txt ...
Student 2: Peter Yang
Course: Studio Art and Theory
store in record.txt ...
Student 3: Wendy Nickeson
Course: Basic Electrical Circuits
store in record.txt ...
```

Figure_2 Sample output of Part I on console (monitor)



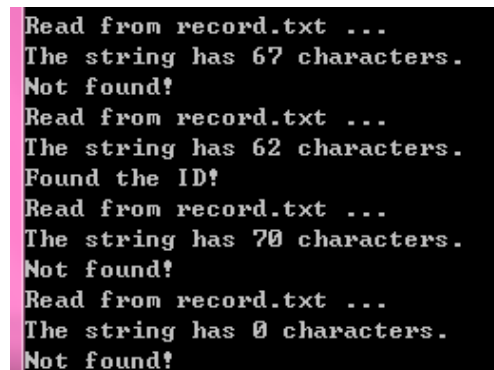
```
record.txt - Notepad
File Edit Format View Help
Name: Fabian MacAdams; ID: 294; Course name: Introduction to Greek
Name: Peter Yang; ID: 518; Course name: Studio Art and Theory
Name: Wendy Nickeson; ID: 376; Course name: Basic Electrical Circuits
```

Figure_3 Sample data of record.txt

Part II (1.5points)

This part will reopen *record.txt* with *input* mode. If it opens successfully, read the line one by one in this file. Then perform below tasks:

1. Display the length of each line;
2. Evaluate if the student ID = 518 is existed in each line; prompt the user a message if it is found or not;
3. Close the file when finishing above operations.



```
Read from record.txt ...
The string has 67 characters.
Not found!
Read from record.txt ...
The string has 62 characters.
Found the ID!
Read from record.txt ...
The string has 70 characters.
Not found!
Read from record.txt ...
The string has 0 characters.
Not found!
```

Figure_3 Sample output of record.txt

A sample output of this program is as following:

```
"D:\Ohlone College\Adjunct\2019 Spring\CS102-04\Lab\Lab7\lab7.exe"
Read from info.txt ...
Student 1: Fabian MacAdams
Course: Introduction to Greek
store in record.txt ...
Student 2: Peter Yang
Course: Studio Art and Theory
store in record.txt ...
Student 3: Wendy Nickeson
Course: Basic Electrical Circuits
store in record.txt ...
Read from record.txt ...
The string has 67 characters.
Not found!
Read from record.txt ...
The string has 62 characters.
Found the ID!
Read from record.txt ...
The string has 70 characters.
Not found!
Read from record.txt ...
The string has 0 characters.
Not found!

Process returned 0 (0x0)   execution time : 517.864 s
Press any key to continue.
```

Requirement:

1. Submit **2** files on Canvas:
 - a. the source file;
 - b. record.txt.
2. NO programmer defined function is required in this program;
3. Your output on the console (monitor) is NOT required to be exactly same as the sample output, you can create different student and course names;
4. The format of student information in record.txt should be same as the sample data, which contains student name, student ID, and course name correspondingly and respectively;
5. Point(s) may be **deducted** if:
 - Missing file(s) or incomplete code, Syntax errors / logic errors;
 - No documentations or comments in the source file;
 - No calling of related functions under string class;
 - No evaluation of open files and/or no close files;
 - Incorrect output.