

# CS150 Assignment 7

## DNA!

**Date assigned:** Monday, November 17, 2008

**Design Documents Due:** Friday, November 21, 2008, 9:15 am (5 points)

**Date due:** Tuesday, December 2, 2008, 5pm (55 points)

**Total points: 60**

### ***Problem***

A DNA sequence is represented as a series of characters: A, C, T, G. In Biology research, scientists often need to study the differences and similarities between multiple DNA sequences. For this assignment, you need to write a C++ program that will allow the user to read two DNA sequences from files and compare and manipulate them in various ways. You must store each DNA sequence as an array of characters. Each DNA sequence will be 16 characters long. The operations that the user needs to be able to perform are:

- Print: print to the screen both of the DNA sequences
- Reverse: reverse each DNA sequence and print each to the screen.
- Pair Finding: allow the user to enter a 2 character DNA fragment and determine if each of the full DNA sequences contains that 2 character fragment. Print an appropriate message to the screen that marks any found pairs.
- Recombine: Combine the two DNA sequences into one new DNA sequence (of length 16). To recombine the DNA, take the first character from the first sequence, the second character from the second sequence, the third character from the first sequence, and so on. The user should be asked for the name of a file to store this new DNA sequence.
- Sums: Count how many of each character (A, T, C, G) each DNA sequence contains and print the results to the screen.

Your program will also need to provide a small menu system, so that the user can choose one of the above operations after inputting the two DNA sequences. The user should be prompted for input until they select a valid menu option. The menu should be redisplayed until the user asks to quit.

If the user specifies an invalid file, you need to ask for a new file name until the file is opened successfully.

## Output

The output for this program should look exactly like this:

```
|+++++|
|PACIFIC UNIVERSITY DNA-ANALYZER|
|+++++|
```

```
Please enter the name of a file containing a DNA sequence: DNA1.txt
Please enter the name of a file containing a DNA sequence: DNA2.txt
```

How would you like to analyze the DNA?

1. Print to screen
  2. Reverse DNA
  3. Pair finding
  4. Sums, count the occurrences of each character
  5. Recombine
  6. Quit
- > 1

Printing...

```
Sequence one: A T T T C G C G A T T T C G C G
Sequence two: C G T A A T G C C G T A A T G C
```

How would you like to analyze the DNA?

1. Print to screen
  2. Reverse DNA
  3. Pair finding
  4. Sums, count the occurrences of each character
  5. Recombine
  6. Quit
- > 2

Reversing ...

```
Sequence one: G C G C T T T A G C G C T T T A
Sequence two: C G T A A T G C C G T A A T G C
```

How would you like to analyze the DNA?

1. Print to screen
  2. Reverse DNA
  3. Pair finding
  4. Sums, count the occurrences of each character
  5. Recombine
  6. Quit
- > 1

Printing...

Sequence one: G C G C T T T A G C G C T T T A  
Sequence two: C G T A A T G C C G T A A T G C

How would you like to analyze the DNA?

1. Print to screen
  2. Reverse DNA
  3. Pair finding
  4. Sums, count the occurrences of each character
  5. Recombine
  6. Quit
- > 3

Pairs...

What two character sequence would you like to look for: T T

Sequence one does contain this pair.

G C G C T T T A G C G C T T T A  
\*\*\*\*\*

Sequence two does not contain this pair.

How would you like to analyze the DNA?

1. Print to screen
  2. Reverse DNA
  3. Pair finding
  4. Sums, count the occurrences of each character
  5. Recombine
  6. Quit
- > 3

Pairs...

What two character sequence would you like to look for: C G

Sequence one does contain this pair.

G C G C T T T A G C G C T T T A  
\*\*\*

Sequence two does contain this pair.

C G T A A T G C C G T A A T G C  
\*\*\*

How would you like to analyze the DNA?

1. Print to screen
2. Reverse DNA
3. Pair finding
4. Sums, count the occurrences of each character

```
5. Recombine
6. Quit
> 4
```

Sums...

```
Sequence one: A: 2 C: 4 T: 6 G: 4
Sequence two: A: 4 C: 4 T: 4 G: 4
```

How would you like to analyze the DNA?

```
1. Print to screen
2. Reverse DNA
3. Pair finding
4. Sums, count the occurrences of each character
5. Recombine
6. Quit
> 5
```

Recombining...

```
New Sequence: G G G A T T T C G G G A T T T C
File name: DNA3.txt
```

How would you like to analyze the DNA?

```
1. Print to screen
2. Reverse DNA
3. Pair finding
4. Sums, count the occurrences of each character
5. Recombine
6. Quit
> 6
```

## Notes

To open a file with a name specified by the user ask the user for a string and use the `c_str()` function to pass the string to the `open()` function.

```
Example
string fileName
ifstream ifile;

// read fileName
ifile.open(fileName.c_str());
```

If you reuse an `ifstream` variable to open a second file, you need to use the `.clear()` function to reset the end of file marker.

```
Example:
ifile.clear();
ifile.open(secondFile.c_str());
```

The two files shown in the example above, `DNA1.txt` and `DNA2.txt`, will be in the **CS150-02 Public** folder on Turing.

## ***To complete this assignment you must***

1. Create a new C++ project in Visual Studio. Name your project "07DNAxxxxxxx", where xxxxxxxx should be replaced by your PU Net Id.
2. Type the solution (fully documented/commented) to the problem into your project.
3. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly. Be aware that C++ is case sensitive.
4. Once you are sure that the program works correctly it is time to submit your program. You do this by logging on to Turing and placing your complete project folder in the CS150-02 drop folder. Make sure that you copy your program folder and don't move it. If you move it, then you will not have your own copy!

## ***Submit an electronic copy of your design document***

Before you start you need to think about the data in your program and the calculations you will need to perform. Answer the following questions in a GoogleDoc (named 07ProgramDesignPUNetID) and share it with the instructor ([profchadd@gmail.com](mailto:profchadd@gmail.com)). Be sure to answer the questions in complete sentences where appropriate. This design document is due on **Friday at 9:15am**.

### **Design Questions:**

1. What data will you need to store for this program, and what are their data types?
2. What functions will you need to write? What will their parameters be and their return types?
3. How many loops will you need for your program and briefly what will each need to do?

### **Notes**

1. You must write and use at least **two functions** in addition to main() in your solution (you may choose to have many more than two functions). One function may have return type of void, the other must return a value.
2. You must use at least one **array**.
3. You must follow the coding standards.
4. You must use constants when possible.
5. Your program will be graded on efficiency. In other words, you will be marked down for repeating code statements unnecessarily.
6. You should easily be able to alter your code to handle a different DNA sequence length.
7. You may only use the C++ programming concepts covered thus far in class. Do not use any more advanced concepts that we have not covered or any other programming concepts that you have had experience with.
8. Your output must look **exactly** like the sample given.
9. You must comment your code appropriately.
10. Refer to the syllabus for what constitutes plagiarism, and the consequences for plagiarizing.

To receive full credit for this assignment, your project must be in the drop box by 5pm on the day that it is due. Anything later will be considered late. Further, you must bring a color hard copy of your program to the instructor by 5pm.

**You may NOT use your gift on this assignment.  
You may NOT submit this assignment late!**

**Good luck! And remember, if you have any problems, come and see me straight away.  
START EARLY!!**