Assignment:

Find $T(n)$, $f(n)$, c, and n_0 for each of the code fragments given in Figures 1, 2, and 3. Put the results of your analysis in your report in the algorithm section. Your analysis should be patterned after the analysis in Figure 8. Transform the code fragment into while-loops. Account for the cost of each line. Sum the total and express it in closed form. Employ the **Microsoft © Equation Editor** to format all mathematical expressions.

Write functions *af01*, *af02*, and *af03* that return $T(n)$ according to your analysis. Functions *af01*, *af02*, and *af03* compute $T(n)$ analytically. Model your design of functions *af01*, *af02*, and *af03* after example analytical function *af00* shown in Figure 6.

Write functions *ef01*, *ef02*, and *ef03* that return $T(n)$ by counting the number of times each statement in the code fragment executes. Functions *ef01*, *ef02*, and *ef03* compute $T(n)$ empirically. Mode your design of functions *ef01*, *ef02*, and *ef03* after example empirical function *ef00* shown in Figure 7.

Put all functions that compute $T(n)$ in file **F09.cpp**. Write function prototypes for each of your functions and put them in file **F09.h.** Write file **p09.cpp** that tests all six functions in file **F09.cpp**. File **p09.cpp** contains function main. Write functions in file **p09.cpp** that read separate files containing values of *n* for each code fragment. Refer to the **input** section to determine which files contain data for functions ef01 and af01; ef02 and af02; and ef03 and af03.

Create file **p09make** containing instructions for the UNIX utility make that compile and link program **p09**.

Prohibition: Use of the C++ Standard Template Library is prohibited in the implementation of this project.

Program Files: Project **9** consists of files **p09.cpp**, **F09.h**, **F09.cpp**, and **p09make**. Project file names are exactly as given. Failure to employ the foregoing names will result in a score of ***zero*** (***0***) for this project

Project files must be stored in the **root directory of your student account**. Failure to store project files in the root directory of your student account will result in a score of ***zero*** (***0***) for this project.

File	Description
p09.cpp	File **p09.cpp** contains functions that process command line arguments and exercise analytical and empirical functions.
F09.h	File **F09.h** contains prototypes for analytical functions *af01*, *af02* and *af03*, and empirical functions *ef01*, *ef02* and *ef03*.
F09.cpp	File **F09.cpp** contains implementations for the functions whose prototypes are defined in file **F09.h.** File **F09.cpp** can contain functions to support computations performed by the analytical and empirical functions contained in the file.
p09make	File **p09make** contains instructions for program **p09**. Instructions are written for the UNIX utility ***make***. Program **p09** is contained in file **p09**.

Command Line: Project **9** can be invoked with zero, one, two, three, or four program parameters. The first three program parameters are the input file names. The fourth parameter is the output file name. Sample command lines together with corresponding actions by program **p09** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

$ **p09**
Enter the first input file name: **i091.dat**
Enter the second input file name: **i092.dat**
Enter the third input file name: **i093.dat**
Enter the output file name: **o09.dat**

$ **p09 i091.dat**
Enter the second input file name: **i092.dat**
Enter the third input file name: **i093.dat**
Enter the output file name: **o09.dat**

$ **p09 i091.dat i092.dat**
Enter the third input file name: **i093.dat**
Enter the output file name: **o09.dat**

$ **p09 i091.dat i092.dat i093.dat**
Enter the output file name: **o09.dat**

$ **p09 i091.dat i092.dat i093.dat o09.dat**

Input files: Files **i091.dat, i092.dat,** and **i093.dat** contain values of n for code fragments (1), (2), and (3) respectively. Program **p09** reads an input file and produces output for that file. Program **p09** reads input files in succession and produces output for each input file as the file is read.

Output files: Send your output to both the output file and the display (stdout). Put your output in three columns. Label the first column **n**, the second column **Analytical** and the third column **Empirical.** Produce values of $T(n)$ for each code fragment. One line is printed for each value of n read. Write the value of n in the column labeled **n**. Write the value of $T(n)$ computed by the designated analytical function, *af0x*, in the column labeled **Analytical**. Write the value of $T(n)$ computed by the selected empirical function, *ef0x*, in the column titled **Empirical**. Produce one table for each code fragment. Identify the tables by the code fragment analyzed. Figure 5 contains the output for example code fragment 0 shown in Figure 4.

Figure 1. Code fragment 1.

```
int sum=1;
for (int a=0;a<n;a++) sum=sum*4;
while (sum>0) sum--;
```

Figure 2. Code fragment 2.

```
for (int i=0;i<n;i++) {
    int m=n;
    while (m>1) m=m/4;
}
```

Figure 3. Code fragment 3.

```
int sum=0;
for(int i=0;i<n;i++)
    for(int j=0;j<i*i;j++)
        for (int k=0;k<j;k++)
            sum++;
```

Figure 4. Example Code Fragment 0

```
int sum=0;
for (int i=0;i<n;i++) sum++;
```

Figure 5. Example Output for Code Fragment 0

n	Analytical	Empirical
1	3	3
10	33	33
20	63	63
30	93	93
40	123	123
50	153	153
60	183	183
70	213	213
80	243	243
90	273	273

Figure 6. Analytical Timing Function af00.

```
int af01(int n)
{  return 3*n+3;
}
```

Figure 7. Empirical Code Function ef00.

```
int ef01(int n)
{  int t=0, i, sum;
   sum=0;                    t++;
   i=0;                      t++;
   while (i<n) {             t++;
     sum++;                  t++;
     i++;                    t++;
   }                         t++;
   return t;
}
```

Figure 8. Analysis of Code Fragment 0

Line	Code	Cost	Reduced Form
1	sum=0;	1	1
2	i=0;	1	1
3	while (i<n) {	$\sum_{i=0}^{n-1} 1$	n
4	sum++;	same as line 3	n
5	i++;	same as line 3	n
6	}	1	1
		$T(n) =$	$3n+3$