Assignment #5

Assigned : Monday October 9, 2017
Due : Tuesday, October 17, 2017 (23:55 Atlantic time).

Hand in source code only. Please add all source code into one big zip file and post that to Moodle, it won't take more than **ONE** file. If it's too large to upload to Moodle, you likely have more than source code. If you are still stuck and can't upload the file, email it to me, cc to yourself and see me in class. Please do not send binary executables, only source code.

Note we can't have an extension on this one as it has to be done in time for the midterm.

Question 1 – SQL (4 pts)

Using Microsoft Access on the lab computers or MySQL or PostgreSQL on the cs.smu.ca, create and fill the following tables. Note that the names are fixed length fields of less than 20 characters and the remaining items are numbers. The maximum allowable Lab Mark is 10, and the maximum allowed Test mark is 100.

tblStudentNo:

studentNo	lastName	firstName
79004651	Somers	Curtis
96000001	Unknown	
96000002	Day	Green
96000003	Oven	Beth
96000004	Lisp	Franz

tblLabMark:

studentNo	labNo	labMark
79004651	1	6
79004651	2	7
96000001	1	6
96000002	2	7
96000003	2	8
96000005	2	7

tblTestMark:

studentNo	testNo	testMark
79004651	1	70
79004651	2	60
96000001	1	50
96000001	2	50
96000002	1	60
96000003	1	70
96000003	2	80
96000002	2	90

Using the above tables, create SQL that will perform the following on your tblStudent. Don't just select the record by number, use the appropriate select, sub-select or SQL function to pick just the ones that answer the following questions:

a. Print the last name, a comma, then first name from concatenated together as one column from the the tblStudentNo table sorted by firstName in ascending order.
b. Show the number of student records in the tblStudentNo table.
c. Select the first and last names of the lowest numbered studentNo in the tblStudentNo table
d. Select the name of the student who did not pass in a lab
e. Select the student firstName, lastName that had the highest lab mark
f. Select the average of lab #2 (limited of course to those four lab #2 actually passed in).
g. Calculate term marks assuming the equation:
   ((Lab#1/10)*15) + ((Lab#2/10))*15) +((Midterm#1/100)*30)+((Midterm#2/100)*40)
   Remember to replace nulls with zeroes.
h. Select the firstName, lastName and final mark of the student with the highest final mark using the equation in g.

If you really don't want to use Access, you can use a Postgres or MySQL database on CS (trust me you really want to use MS Access). For example if you want to use PostgreSQL, ask the cs admin for a username and password and to set you up with a database. Once you have that, type (if you had the username csc43001):

psql –d csc43001 -u csc43001 -p

The system will then prompt for a password. Once in the PostgreSQL system, you can simply type in valid SQL commands and get the system to run them. A quick help shows up when you log on that displays the following:

\h for help with SQL commands
\? for help with psql commands
\g or terminate with semicolon to execute query
\q to quit

Entering \? from the prompt will display the most useful commands. \dt can be used to list the tables in your database. There are several good tutorials available on the use of PostgreSQL. Best source of reference is probably the PostgreSQL home pages at http://www.postgresql.org/

Remember that a missing value is equal to zero, and you can use sub-selects (nested selects) and can add any valid functions that you wish but please don't call SQL from something like a C++ or VB application. You can submit queries but it is easier to submit the whole database if you are doing this in MS Access or SQL Server. If you do this assignment in PostgresSQL or MySQL, please just submit the SQL, not the whole database.

Question 2 – C and Pascal (6 pts)

Write a **RECURSIVE** factorial program in both **C (not C++) and Pascal**. You can use the gnu Pascal (gpc) or free pascal compiler (fpc) and the gnu C compiler. Try and use the least number of lines of code in each language and say which one has the least number? If you are on the cs machine, please use the GNU Pascal Compiler "gpc" (http://www.gnu-pascal.de/gpc/h-index.html). If you are on your own system, you can try and install the free pascal compiler "fpc" (http://www.freepascal.org/). If you are using anything else, please let me know.

2. (2 pt) Convert the following Pascal program into C (not C++). Using comments in the code, Identify at least 4 places where the code is shorter in Pascal or C.

```
program tryit (input, output);
   var i, n: integer;
```

```
begin
   n := 0;
   writeln('Enter an integer:');
   readln(n);                        { read an input integer}
   for i := n downto 1 do
      writeln(i*i*i)
   writeln('Well that was easy!');
end.
```

If you want to see what that code does, copy it to the cs computer and compile it using Gnu Pascal Compiler (gpc) or free Pascal Compiler (fpc) on the cs computer.

3. (6 pts) Write the equivalent C program translation (NOT C++) for the following Pascal program. The program must compile using gcc (not g++) on the cs machine. Note the use of nested functions for min and swap. This can be translated directly since nested functions are now legal in C. Note "sArray" is passed into the functions by ref. Your C code should hand them by ref as well. A good reference is the Wikipedia Page - Comparison of Pascal and C.

```
{*****************************************
       Sample Pascal program
*****************************************}
program linearSearch(input,output);

type
       searchArrayType = array [1..20] of real;

procedure Introduction;
begin {Introduction}
       writeln('This program reads numbers from a file called input.dat,');
       writeln('sorts the numbers.');
end; {Introduction}

procedure sort(var store: searchArrayType);
       {If you moved min out of this procedure, you will have to pass store}
       function min(start: integer): integer;
       var
              i: integer;
              tempMin: integer;
       begin {min}
              tempmin := start; {Note: case insensitive}
              for i := start+1 to 20 do
                     if store[i] < store[tempMin] then
                            tempMin := i;
              min := tempMin;
       end; {min}

       procedure swap(var l,r : real);
       var
              temp: real;
       begin {swap}
              temp := l;
              l := r;
              r := temp;
       end; {swap}
var
       i: integer;
begin {sort}
       for i := 1 to 19 do
              swap(store[i],store[min(i)]);
end; {sort}

procedure display(store: searchArrayType);
var
```

```
       i: integer;
begin {display}
        for i := 1 to 20 do
             writeln(store[i]:7:2);
end; {display}

procedure readInput(var store: searchArrayType);
var
       infile : TEXT;
       i : integer;
       filename: String[100];
begin {readInput}
       write('Please input the filename: ');
       readln(filename);
       assign(infile,filename);
       reset(infile);
       for i := 1 to 20 do
       begin
             read(infile,store[i]);
       end;
end; {readInput}

var
       sArray: searchArrayType;
begin {main}
       introduction;
       readInput(sArray);
       sort(sArray);
       writeln('Sorted numbers are as follows:');
       display(sArry);
end. {main}
```