Take Home C Exam for Physics 4051  
Due November 27, 2002 at the beginning of class  
25 points – Work on your own!* 

Calculate the average temperature for a given month and year from the data given in the additional exercise 11.2. This assignment consists of two parts.

First, write a self-contained, ANSI C function that will return the average temperature for a specific month and year. Specifically, your function will search through all the average daily temperatures records read in by the function ReadTemp and it will average the ones that fit a specific month number and year. In case there are less than 25 records available for the specified month and year then the function must return −999. See the additional information below.

Second, write a complete program that uses the above function to determine and display the average monthly temperature (if available) each time the user enters a specific month number and year. If the function returns −999, display a message indicating that data for this month and year is not available. This program can be written either in ANSI C or LabWindows. The temperature data is stored in the data file named WeatherFile.csv in the folder U:\pub\LW\F02. As in exercise 11.2, you will again use the ReadTemp function to read the data into arrays, but this time you will use the average daily temperature values to determine the monthly average. See the additional information below.

Additional Information and Hints:

First Part:
A self-contained ANSI-C function means that it does not rely on any external global variables! Therefore, you are only allowed to use local variables and values passed through the function arguments.

Your function may not call the function ReadTemp; instead, you must pass all required values read in by ReadTemp to your function through your function’s arguments.

You must write your own averaging algorithm. You are not allowed to use LabWindows (or ANSI-C) sorting or analysis functions such as “Mean” or “Sum1D” though you may want to use them for debugging and testing.

Finally, all values read by ReadTemp are in degrees Celsius. Your function may return the result in degrees Celsius or Fahrenheit.

Second Part:
Information on reading the data file named WeatherFile.csv from the folder U:\pub\LW\F02 was given in the previous handout for Exercise 11.2, but it can also be obtained from the web at: http://mxp.physics.umn.edu/f02/Assignments/AveTemp3a.htm
(You can copy/paste the function ReadTemp directly into your code by opening the file ReadTemp.c in the folder U:\pub\LW\F02 with LabWindows.)

*General:
For this assignment you may consult any book or the lecture notes. If you have additional questions see Kurt Wick or Prof. Crowell. You may not ask your fellow students or TA for help. If you cannot do both parts write either part of the program; you may receive partial credit.
ANSI-C Solution:

#include <ansi_c.h>
define MAXPTS 1000

short ReadTemp( int *yr, int *mth, int *dy, float* tmp, float* mx, float* mn);
double MonthlyAve(int *yr, int *mth, float* tmp, int n, int mthsel, int ysel);

main()
{
    int iread, imonthsel, iyearsel;
    static float Tav[MAXPTS], fgarb[MAXPTS];
    static int igarb[MAXPTS], iMonth[MAXPTS], iYr[MAXPTS];
    double dmav;

    printf("Enter Month Number: "); //get user input for month
    fflush( stdin );
    scanf("%i", &imonthsel);
    printf("Enter Year: ");     //and year
    fflush( stdin );
    scanf("%i", &iyearsel);

    //read in all the temperature data and store it in the arrays
    iread = ReadTemp( iYr, iMonth, igarb, Tav, fgarb, fgarb);

    //call the averaging function with the data
    dmav = MonthlyAve(iYr, iMonth, Tav, iread, imonthsel, iyearsel);
    if(  dmav > -999.0)  //if data is good print result
        printf("Average Temp in C: %f for Month: %d Year: %d\n", dmav, imonthsel, iyearsel);
    else
        printf("No data available for Month: %d Year: %d\n", imonthsel, iyearsel);
}

//Function calculates and returns the monthly average temperature for a
//specific month (mthsel) and year (ysel)
double MonthlyAve(int *yr, int *mth, float* tmp, int n, int mthsel, int ysel)
{
    int i, icount = 0;
    double dtot = 0;

    //sum the temperatures for a particular month and year
    for( i = 0; i < n; i++)
        if( (yr[i]==ysel) && (mthsel==mth[i]) ) //only sum if above condition
            {           //is true
                dtot += tmp[i];
                icount++;  //count the number of days summed for that month
            }

    if( icount > 25)    //check if enough data available for that month
        return( dtot/icount);  //if so, return average
    else
        return(-999.0);    //return an error code
}

Not shown here: Function ReadTemp below....