

Sep 14, 20 20:14

main.c

Page 1/3

```

/*
Assignment #1: C Programming Basics Part-1
Enter 5 angles by user input or data file
Return results in a soln file with date and time stamp

Author: Shawn Hinnebusch

Date: 9/14/2020

To compile: gcc -o hw1.exe main.c -lm
Example input:
0
10,20,30,40,50

or
1
"name of file to import"

width: 7
precision: 4
*/

#include <math.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define SIZEOF(a) sizeof(a) / sizeof(*a)
#ifndef M_PI
#define M_PI 3.1415926535897932
#endif

void getDateForFileName(char *resultsFile);

int main()
{
    double inputAngle[ 5 ];
    double solnAngle[ 5 ];
    double convert;
    int enterImport;

    // Allow user to select enter data or import it
    printf("Enter 0 to enter data or 1 to import a text file: ");
    scanf("%d", &enterImport);
    printf("\n");

    if (enterImport == 0) {
        printf("Enter five values for angles in degrees separated by a comma: \n");
        scanf("%lf,%lf,%lf,%lf,%lf", &inputAngle[ 0 ], &inputAngle[ 1 ], &inputAngle[
2 ], &inputAngle[ 3 ],
        &inputAngle[ 4 ]);
    } else if (enterImport == 1) {
        char inputFileName[ 30 ];
        printf("Enter input file name: ");
        scanf("%s", inputFileName);

        FILE *input;
        input = fopen(inputFileName, "r");

        if (input == NULL) {
            printf("Error Reading File\n");
            return -1;
        }

        for (int n = 0; n < SIZEOF(inputAngle); n++) {
            fscanf(input, "%lf,", &inputAngle[ n ]);
        }
    }
}

```

Sep 14, 20 20:14

main.c

Page 2/3

```

        fclose(input);
    } else {
        printf("Invalid input\n");
        return -1;
    }

    for (int n = 0; n < SIZEOF(inputAngle); n++) {
        if (!(inputAngle[ n ] <= 360 && inputAngle[ n ] >= 0)) {
            printf("at least one of the input values is outside of the allowed interval [0,360]\n");
            return -1;
        }
    }

    for (int n = 0; n < SIZEOF(inputAngle); n++) {
        convert = sqrt(fabs(cos(inputAngle[ n ] * M_PI / 180.0)));
        solnAngle[ n ] = log10(convert) / log10(2.0);
    }

    // Get date/time and combine into 1 variable
    char resultsFile[ 50 ] = "soln_";
    getDateForFileName(resultsFile);

    // Allow the user to specify the desired width and precision
    char width[ 5 ];
    printf("Enter Desired Width: ");
    scanf("%s", width);
    char formatSettings[ 10 ] = "%";
    strcat(formatSettings, width);
    strcat(formatSettings, ".");
    char digits[ 5 ];
    printf("Enter Desired precision: ");
    scanf("%s", digits);
    strcat(formatSettings, digits);
    strcat(formatSettings, "f\n");

    // Output the results to the file
    FILE *output;
    output = fopen(resultsFile, "w");

    if (!output) return 1;
    for (int n = 0; n < SIZEOF(inputAngle); n++) {
        fprintf(output, formatSettings, solnAngle[ n ]);
    }

    fclose(output);

    return 0;
}

// Functions
void getDateForFileName(char *resultsFile)
{
    // Obtain current time
    int hours, minutes, seconds, day, month, year;
    time_t now;
    time(&now);
    struct tm *local = localtime(&now);

    hours = local->tm_hour; // get hours since midnight (0-23)
    minutes = local->tm_min; // get minutes passed after the hour (0-59)
    seconds = local->tm_sec; // get seconds passed after the minute (0-5
9)
    day = local->tm_mday; // get day of month (1 to 31)
    month = local->tm_mon + 1; // get month of year (0 to 11)
    year = local->tm_year + 1900; // get year since 1900

    // Convert int to string
}

```

Sep 14, 20 20:14

main.c

Page 3/3

```
char cday[ 5 ], cmonth[ 5 ], cyear[ 5 ], chours[ 5 ], cminutes[ 5 ], cseconds[ 5 ];
sprintf(cday, "%d", day);
sprintf(cmonth, "%d", month);
sprintf(cyear, "%d", year);
sprintf(chours, "%d", hours);
sprintf(cminutes, "%d", minutes);
sprintf(cseconds, "%d", seconds);

// Combine date and time into the correct formate in 1 string
strcat(resultsFile, cmonth);
strcat(resultsFile, "-");
strcat(resultsFile, cday);
strcat(resultsFile, "-");
strcat(resultsFile, cyear);
strcat(resultsFile, "-");
strcat(resultsFile, chours);
strcat(resultsFile, ":");
strcat(resultsFile, cminutes);
strcat(resultsFile, ":");
strcat(resultsFile, cseconds);
strcat(resultsFile, ".dat");
};
```