

Homework #5 – Solution

(C Programming for Beginners - OnLine)

Note: If you complete this (specially 5.1 and 5.2) and next homework, you will be much closure to your in designing your midterm project

5.1 Write a program, which will display the menu with 5 items. It will ask the use to give their choice. It will trap all possible errors and then print the result when correct input is provided before exiting the program.

Example user interface:

```
Welcome to sorting program
```

1. Title
2. Rank
3. Date
4. Stars
5. Likes

```
Enter your choice between 1 and 5 only: s;fa fa f
```

```
You have entered an invalid choice. Try again.
```

```
Enter your choice between 1 and 5 only: 9
```

```
You have not entered a number between 1 and 5. Try again.
```

```
Enter your choice between 1 and 5 only: 3
```

```
You entered valid choice 3
```

```
Thank you for giving your choice
```

Solution:

```
/******
```

```
Program: GetMenuChoice
```

```
Description: This program gives the user menu choices  
between 1 – 5 items and asks the user to select one of them. If  
they enter a number outside of that boundary, or they enter  
characters then it will give an alert and ask the user to enter  
correct choice.
```

```
It continues until user enters correct number
```

```
*****/
```

```
#include <stdio.h>
```

```

/*****
Function: main
Description: Displays the menu choice and goes in a loop to ask
the user their choice. If they enter wrong choice gives
appropriate error message and asks the user to re-enter
Once right choice is given, it returns that value to calling
function

input: not used

return: not used

*****/

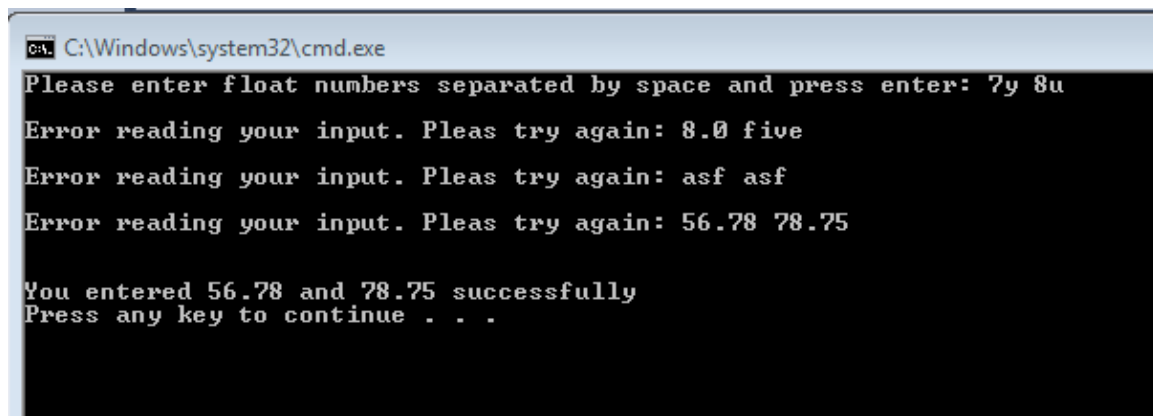
int main(int argc, const char * argv[]) {
    int nChoice = 6; //initiatlize with an invalid choice

    printf("Sorting Menu:\n\n\t1. Rank\n\t2. Title\n\t3. Date\n\t4. Stars\n\t5. Likes\n\n");
    //scanf returns number of successful translation.
    //If user inputs characters instead of numbers it will not
    //return 1 as you are scanning
    //one value. This is the way to trap wrong user input
    while (nChoice < 1 || nChoice > 5)
    {
        printf ("Enter your choice between 1 and 5 only: ");
        while ( (scanf("%d", &nChoice)) != 1)
            //if the scan was correct scanf would return 1
        {
            //you come here when there is wrong input from user
            //and there may be garbage characters inputted by
            //user, eat it up until
            //input buffer is clear indicated by newline
            while (getchar() != '\n')
                continue;
            printf("\nYou have entered an invalid choice. Try
                again.");
        }
        if (nChoice < 1 || nChoice > 5)
            printf("You have not entered a number between 1 and
                5. Try again.\n");
    }
    //if you come here, that means the choice given was correct.
    printf("\nYou entered valid choice %d\n", nChoice);
    printf("Thank you for giving your choice\n");
}

```

5.2 Write a program that will give the user prompt to enter two float (or double) values. If the values inputted are correct then display the inputted two values. If user enters characters instead of numbers or if they enter invalid numbers then the program will display the error message and ask the user to re-enter the correct values again. It only exits when the correct input is received and displayed.

Example user interface:



```
C:\Windows\system32\cmd.exe
Please enter float numbers separated by space and press enter: 7y 8u
Error reading your input. Please try again: 8.0 five
Error reading your input. Please try again: asf asf
Error reading your input. Please try again: 56.78 78.75

You entered 56.78 and 78.75 successfully
Press any key to continue . . .
```

Solution:

```

/*****
Program: gettwofloats
Description: This program gives the user prompt to enter two
float values.
If the values inputted are correct then two values are printed.
If user enters characters instead of numbers or if they enter
only one number then the program will display the error message
and ask user to enter again.

Revision 1: 2/11/2016

Known Issue: It does not validate ridiculous float values –
meaning out of ranges.
Also, user can enter one number and then press enter and then
they can enter
another number and press enter. If user enters two correct
values in the
beginning and garbage values after that, scanf ignores those
garbage values

*****/
```

```

//all includes
#include <stdio.h>

/*****
Function: main
Description: This function asks the user to enter two floats and
then verifies that the numbers are valid by checking return from
scanf which returns the number of successful scans. Since we
need two successful scans, we can simply test for number 2.

If error, it will first clear the buffer which may have garbage
values and we don't need those (meaning flushes the buffer), and
then prompts the user to re-enter

input:      input not used

output:     displays the two float values entered by user
*****/

int main(int argc, const char * argv[]) {
    float fNums1;
    float fNums2;

    printf("Please enter two float numbers separated by a space
           and press enter: ");
    //scanf returns number of successful translation. If user
    //inputs characters instead of numbers it will not return 2.
    //This is the way to trap wrong user input
    while ( (scanf("%f %f", &fNums1, &fNums2)) != 2)
    {
        //you come here when there is wrong input from user and
        //there may be garbage characters inputted by the user,
        //eat up each character until buffer is clear
        //indicated by a newline
        while (getchar() != '\n')
            continue;
        printf("\nError reading your input. Pleas try again: ");
    }

    //you come here only when user enters two correct numbers
    //(floats) you are done just return.

    printf("\n\nYou entered %5.2f and %5.2f successfully\n",
           fNums1, fNums2);

    return 0;
}

```

5.3 Here is a definition of integer array for 7 temperatures of the week. It is already initialized and there are variables declared for the minimum, maximum, total and average temperatures. These variables are going to be used to get the these values for this weekly temperature

```
int weeklyTemp[] = { 69, 70, 71, 68, 66, 71, 70 };
int i, max = 0, min = 0;
float total = 0, average;
```

Given the above definition, write the code to:

- Print the daily temperature (assume, first day Sunday), e.g.
The temperature on day 1 was 69:
- Find and print the minimum and maximum temperature of the week
- Calculate and print the average temperature of the week

Hint: Use loops (for or while)

Sample Output:

```
The temperature on day 1 was 69:
The temperature on day 2 was 70:
The temperature on day 3 was 71:
The temperature on day 4 was 68:
The temperature on day 5 was 66:
The temperature on day 6 was 71:
The temperature on day 7 was 70:
```

```
The Minimum temperature is: 66
The Maximum temperature is: 71
The average temperage for the week is: 69.28571
```

Thank you for using my homework #6 solution

Solution:

```
printf("\n\nYou entered %5.2f and %5.2f successfully\n",
      fNums1, fNums2);

printf("\nThank you for using my program\n");

int weeklyTemp[] = { 69, 70, 71, 68, 66, 71, 70 };
int i, max = 0, min = 0;
float total = 0, average;
```

```
// print temperatures
for (i = 0; i < 7; i++) {
    printf("\nThe temperature on day %d was %d: ",
        i + 1, weeklyTemp[i]);
}
printf("\n\n");

// find the max, min temperature
for (i = 0; i < 7; i++) {
    if (i == 0)
        max = min = weeklyTemp[i];
    if (weeklyTemp[i] > max)
        max = weeklyTemp[i];
    if (weeklyTemp[i] < min)
        min = weeklyTemp[i];
}
printf("The Minimum temperature is: %d\n", min);
printf("The Maximum temperature is: %d\n", max);

// get average
for (i = 0; i < 7; i++)
    total += weeklyTemp[i];

average = total / 7;
printf("The average temperage for the week is: %5.2f", average);
printf("\nThank you for using my homework #6 solution");
```