Homework #6 – Solution
(C Programming for Beginners - OnLine)

Note: *If you complete this (specially 6.1 and 6.2) and next homework, you will be much closure to your in designing your midterm project*

1. Write a function called `getMenuChoice` (), which returns an integer as menu choice based on the requirement in the problem #1 of "Homework #5". You can utilize the code in the problem #1 for this function. Modify the main function and call your new function `getMenuChoice` to get the choice from user. Print the choice given by the user in the main function.

```
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:

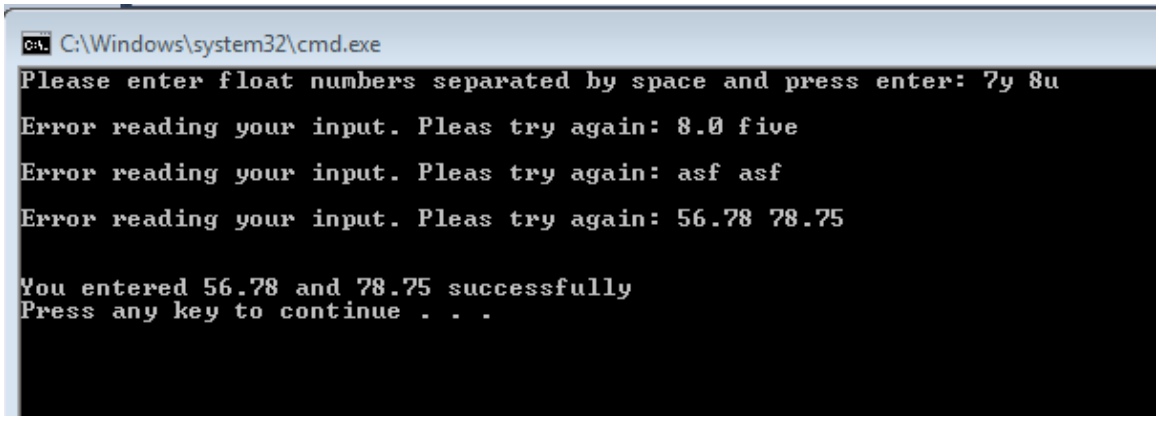
```
/*****************************************************************
Program: GetMenuChoice

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: getMenuChoice

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: none

return: valid choice entered by the user
*****************************************************************/

int getMenuChoice() {

    int nChoice = 6; //initiatlize with a wrong 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 ("Please enter your sorting choice (between 1-5):
              ");
     while ( (scanf("%d", &nChoice)) != 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("\nError reading your input. Pleas try again:
                  ");
     }
     if (nChoice < 1 || nChoice > 5)
           printf("Choice %d is not valid, please try again\n",
                 nChoice);
    }
    //if you come here, that means the choice given was correct.
    return nChoice;
}
```

```
/*****************************************************************
Name: main

Description: Simply calls the getMenuChoice to get the valid
choice from user and then displays that choice on the screen

input: none
return: zero
*****************************************************************/

int main ()
{
    int usersChoice;
    usersChoice = getMenuChoice();
    printf("\n\nYou entered correct choice: %d \n", usersChoice);

    return 0;
}
```

2. Write a function called getTwoFloats(), which returns an array of two floats based on the requirement in the problem #2 of "Homework#5". You can utilize the same code in the problem #2 in this function. Modify the main function to call getTwoFloats and print the two floats in the main method.

Example user interface:



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: 7/28/2015

Known Issue: It does not validate ridiculous float values --
meaning out of float 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: getTwoFloats

Description: This method 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 gobble up 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:      fNums[] Empty array of two floats

output:     fNums[] filled array with two valid floats
*****************************************************************
************/

void getTwoFloats(float fNums[])
{
    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", &fNums[0], &fNums[1])) != 2)
    {
     //you come here when there is wrong input from user and
     //there may be garbage characters inputed by user, eat it up
     //until buffer is clear indicated by newline
     while (getchar() != '\n')
          continue;
     printf("\nError reading your input. Pleas try again: ");
    }
```

```
    //you come here only when user enters
    //two correct nimbers (floats)

    //you are done just return.

    //you may wonder, how does the value get returned to calling
    //function?
    //Since array name itself is the address, so, it behaves like
    //a pointer, you don't have to return the value, the array
    //will reflect the new value in the calling routine
}

/*****************************************************************
Name: main
Description: Declares array of two floats and passes it to
getTwoFloats to get two valid floats.  Once returned, displays
the numbers in the screen

input: none

return: zero
*****************************************************************/

int main ()
{
    float myFloats[2]=  {0.0, 0.0}; //initialize to zero

    getTwoFloats(myFloats);
    printf("\n\nYou entered %5.2f and %5.2f successfully\n",
           myFloats[0], myFloats[1]);
    return 0;
}
```