```
Script started on 2021-10-02 22:02:04-05:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS=
k_gandhi6@ares:~$ cat OSWTA.info
Name: Kush Gandhi

Class: CSC122-001

Levels: 2

Title: Oops shall we try again (Lab)

Description: Testing all types of values when user enters them

by the use functions and a libary.

k_gandhi6@ares:~$ cat InputValidation.h
#ifndef INPUT_VALIDATION_H_INC

#define INPUT_VALIDATION_H_INC



#include <vector>

#include <string>

using namespace std;



double Double_both_Bound(double lowBound, double upBound, string prompt);

long Long_Both_Bound(long lowBound, long upBound, string prompt);

long Long_Low(long lowBound, string prompt);

long Long_Up(long upBound, string prompt);

long Num_Check_Long(vector<long> &numbers, string prompt);

double Num_Check_Double(vector<double> &numbers, string prompt);

char Char_low_bound(char lowBound, string prompt);

char Char_Up_Bound(char UpBound, string prompt);

double Double_LowBound(double lowBound, string prompt);

double Double_UpBound(double upBound, string prompt);

#endif

k_gandhi6@ares:~$ cat OSWTA.cpp
#include <iostream>
#include <vector>

#include <cctype>

#include <string>

#include "InputValidation.h"

using namespace std;



double Double_both_Bound(double lowBound, double upBound, string prompt) {

    double num;



    do {

        cout << endl << prompt;

        cin >> num;

    } while (num <= lowBound || num >= upBound);

    return num;

}



long Long_Both_Bound(long lowBound, long upBound, string prompt) {

    long num;

    do {

        cout << endl << prompt;

        cin >> num;

    } while (num <= lowBound || num >= upBound);

    return num;

}



long Long_Low(long lowBound, string prompt) {

    long num;
    do {
        cout << endl << prompt;
        cin >> num;
    } while (num < lowBound);
    return num;
}

long Long_Up(long upBound, string prompt) {
    long num;

    do {
        cout << endl << prompt;
        cin >> num;
    } while (num > upBound);
    return num;
}

long Num_Check_Long(vector<long> &numbers, string prompt) {
    long num;
    bool end = false;

    for (auto i : numbers) {//loop to print nums from vector
        cout << i << " ";
    }
    do {
        cout << prompt;
        cin >> num;

        for (vector<long>::size_type i = 0; i < numbers.size(); i++) {
            if (numbers[i] <= num) {
                end = true;
            }
        }
    } while (cin.fail() || !end);
    return num;
}

double Num_Check_Double(vector<double> &numbers, string prompt) {
    double num;
    bool end = false;

    for (auto i : numbers) {//loop to print nums from vector
        cout << i << " ";
    }
    do {
        cout << prompt;
        cin >> num;

        for (vector<double>::size_type i = 0; i < numbers.size(); i++) {
            if (numbers[i] <= num) {
                end = true;
            }
        }
    } while (cin.fail() || !end);
    return num;
}

char Char_low_bound(char lowBound, string prompt) {
    char value;
    do {
        cout << endl << prompt;
        cin >> value;
    } while (value < lowBound);
    return value;
}

char Char_Up_Bound(char UpBound, string prompt) {
    char value;
    do {
        cout << endl << prompt;
        cin >> value;
    } while (value > UpBound);
    return value;
}

double Double_LowBound(double lowBound, string prompt) {
    double num;

    do {
        cout << endl << prompt;
```

```
        cin >> num;
    } while (num < lowBound);
    return num;
}

double Double_UpBound(double upBound, string prompt) {
    double num;

    do {
        cout << endl << prompt;
        cin >> num;
    } while (num > upBound);
    return num;
}

int main() {
    vector<long> choiceLong;
    choiceLong.push_back(1);
    choiceLong.push_back(3);
    choiceLong.push_back(9);
    choiceLong.push_back(10);
    choiceLong.push_back(24);
    choiceLong.push_back(48);
    choiceLong.push_back(60);
    choiceLong.push_back(100);

    vector<double> choiceDouble;
    choiceDouble.push_back(3.3);

    choiceDouble.push_back(7.5);

    choiceDouble.push_back(65.1);

    choiceDouble.push_back(10.4);

    choiceDouble.push_back(19.8);

    choiceDouble.push_back(65.2);

    choiceDouble.push_back(55.5);

    choiceDouble.push_back(14.6);

    cout << Double_both_Bound(10.2, 125.8

        , "Enter a number between 10.2-125.8: ") << "\n";

    cout << Long_Both_Bound(100, 5000

        , "Enter a number between 100-5000: ") << "\n";

    cout << Double_UpBound(50.6, "Enter a number < 50.6: ") << "\n";

    cout << Double_LowBound(40.4, "Enter a number > 40.4: ") << "\n";

    cout << Num_Check_Long(choiceLong, "Choose from the list: ") << "\n";

    cout << Num_Check_Double(choiceDouble, "Choose from the list: ") << "\n";

    cout << Long_Low(70, "Enter a number > 70: ") << "\n";

    cout << Long_Up(2000, "Enter a number < 2000: ") << "\n";

    cout << Char_low_bound('D', "Enter a character less than D: ") << "\n";

    cout << Char_Up_Bound('k', "Enter a character greater than k: ") << "\n";

}

k_gandhi6@ares:~$ CPP OSWTA
OSWTA.cpp***


k_gandhi6@ares:~$ ./OSWTA.out

Enter a number between 10.2-125.8: 5.4

Enter a number between 10.2-125.8: 12.5
12.5

Enter a number between 100-5000: 55

Enter a number between 100-5000: 101
101

Enter a number < 50.6: 60.5

Enter a number < 50.6: 20.6
20.6

Enter a number > 40.4: 30.5

Enter a number > 40.4: 50.4
50.4
1 3 9 10 24 48 60 100 Choose from the list: 3
3
3.3 7.5 65.1 10.4 19.8 65.2 55.5 14.6 Choose from the list: 3.3
3.3

Enter a number > 70: 54

Enter a number > 70: 90
90

Enter a number < 2000: 6000

Enter a number < 2000: 60
60

Enter a character less than D: a
a

Enter a character greater than k: Z
Z
k_gandhi6@ares:~$ cat OSWTA.tpq
1. How can you pass a prompt or error message

to a function as an argument?



 There are 2 ways to pass it. By passing the return

 error or return the number.



2. How do you pass a string to a function?

Will the strings need to be changed here?

What care do you need to take for these arguments, then?
```

We could use parameters in our functions instead of passing it.

No, the strings will not need to be changed because the user descides how to write them.

3. How do you pass a list of values to a function?
(Warning! There could be two [related] answers!)
Will the elements need to be changed here?
What care do you need to take for these arguments, then?

You could use a list by arrays or vectors.
I used vectors to make a list of the nummbers.
The elements can't be changed and they protected call by refrence.

4. What other arguments does each function take?
Are they changed? What special care should you take with them?

Arguments can be passed by const by refrence to ensure they are not altered. Some of my arguments used are char, double, long, vectors, and int.

5.What value is returned by your functions?
What type is it and what does it represent?

My return values are the variables that reads the user response and returns them. Most of it are long, char, and double return values followed by varaibles.

6.What care does a caller of your functions have to take with this return value?
(Can they immediately assume it is a valid entry?)

If the user enter an inncorrect number or character, the function will repeats itself untill the user enters it correctly. Once it passes to another statement then they will know if it's valid.

7.How does the compiler distinguish which of your functions is being used for a particular call?
(They ALLhave the same name, after all...)

The compiler descides on the parameters used to make our functions. For exmaple, in a driver test, we can write the specefic needs for a function to work which the compiler will find in the .cpp file.

8.How do you protect your library from being circularly included?

When we use #ifndef and #define followed by #endif at the end of a libary. So, the compiler will know what to do with the libary by including the 3 lines of code.

9. What changes are needed in your main application
(the test application here) to get it to work with the library?

```
What about the compiling process?


 We would have to include the libary that we created in order for

 the program to work. When we compile the program it should all do it

 at the same time.


10. How many files does your library consist of? What are they?

Which one(s) do you #include?


 I have 2 files which are the implementation and interface files.

 The implementation file consist of functions of how the program works.

 The interface file consist of the prototypes of the functions.

 Lastly, all I did was #include "InputValidation.h" in to main

 file(implementation file).

k_gandhi6@ares:~$ exit
exit

Script done on 2021-10-02 22:04:31-05:00 [COMMAND_EXIT_CODE="0"]
```