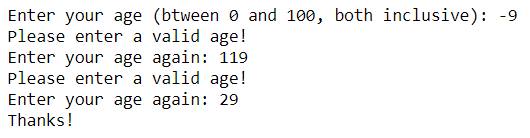
**Week 9 In-Class Exercises (while-loops)**

**Q1: Input Validation**

We often need to use while-loops to validate a user’s input. This exercise is all about validating user inputs.

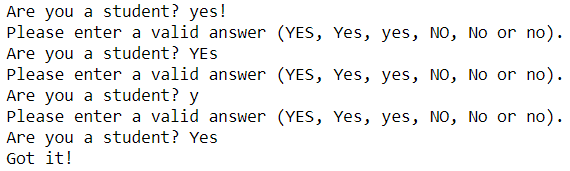
1. [ \* ] Prompt the user for his/her age. Keep prompting the user ***until the input is between 0 and 100*** (both inclusive).

A sample run of the program is shown below:



1. [ \* ] Ask the user whether he/she is a student. If the answer from the user is ***neither “yes” nor “no”***, keep prompting the user until the answer is valid. The program should be flexible enough to accept “YES”, “Yes” and “yes” (but not “yEs”, “YEs”, etc.). Similarly for “no”.

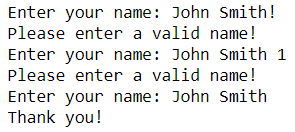
A sample run of the program is shown below:



1. [ \*\* ] Prompt the user for his/her name. The program should only accept a name that ***consists of only letters and spaces***. If the string entered by the user does not satisfy this condition, keep prompting the user until the conditions are satisfied. (You can assume that the user’s input is not an empty string.)

**Hint:** First define a function that checks if a string is considered a valid name.

A sample run of the program is shown below:



**Q2: Integers [\*\*]**

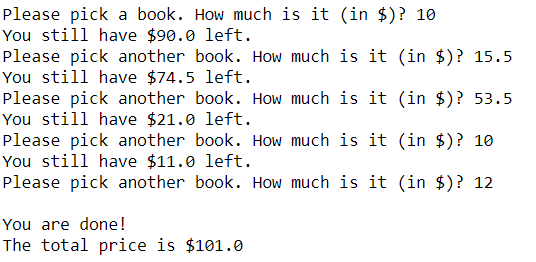
Prompt the user for a sequence of integers ***until the user enters a negative integer***. (You can assume that the user always enters an integer.) The program then displays (in one row) all the odd positive integer that the user has been entered, separated by commas.

A sample run of the code can be found below:

Enter an integer: 7  
Enter an integer: 10  
Enter an integer: 51  
Enter an integer: 93  
Enter an integer: 0  
Enter an integer: -9  
  
Thanks!  
The odd positive integers you have entered are the following:  
7, 51, 93

**Q3: Voucher [ \*\* ]**

You have a $100 voucher to spend in a bookstore. Because the voucher has to be spent on a single receipt, you want to buy at least $100 of books and you are willing to top up any additional cost. Write a program that helps you check whether the books you’ve picked are enough to reach/exceed $100. The program should keep prompting you to buy one more book until the total price is ***equal to or greater than*** $100. A sample run of the program can be found below:



**Q4: Reset PIN**

**Part (a) [ \*\* ]**

Write a program that allows a user to reset his/her PIN number. The program keeps prompting the user for a new PIN number twice (the second time for confirmation) until the following conditions are satisfied:

* The user has entered the same PIN number twice.
* The entered PIN number is a string with 6 digits.

A sample run of the program can be found below:

Enter your new PIN: ***123456***  
Confirm your new PIN: ***12345***  
Sorry! There is an error!  
  
Enter your new PIN: ***1234567***  
Confirm your new PIN: ***1234567***  
Sorry! There is an error!  
  
Enter your new PIN: ***12345a***  
Confirm your new PIN: ***12345a***  
Sorry! There is an error!  
  
Enter your new PIN: ***123456***  
Confirm your new PIN: ***123456***  
  
Thanks! Your new PIN has been set!

**Part (b) [ \*\* ]**

Improve the program above such that an appropriate error message is printed. The error message should include ***all*** the errors found in the two PIN numbers entered. You only need to report the following errors:

* The PIN number is too long (based on the first PIN number entered).
* The PIN number is too short (based on the first PIN number entered).
* The PIN number contains a non-digit character (based on the first PIN number entered).
* The second PIN doesn’t match the first PIN.

A sample run of the program can be found below:

Enter your new PIN: ***123456***  
Confirm your new PIN: ***12345***  
Sorry! The following errors are detected:  
 - The second PIN doesn't match the first PIN.  
  
Enter your new PIN: ***1234567***  
Confirm your new PIN: ***1234567***  
Sorry! The following errors are detected:  
 - The PIN number is too long.  
  
Enter your new PIN: ***12345a***  
Confirm your new PIN: ***12345a***  
Sorry! The following errors are detected:  
 - The PIN number contains a non-digit character.  
  
Enter your new PIN: ***345a***  
Confirm your new PIN: ***12345***  
Sorry! The following errors are detected:  
 - The PIN number is too short.  
 - The PIN number contains a non-digit character.  
 - The second PIN doesn't match the first PIN.  
  
Enter your new PIN: ***123456***  
Confirm your new PIN: ***123456***  
  
Thanks! Your new PIN has been set!

**Q5: Strings (\*\*\*)**

Define a function called get\_strings\_with\_digits() that takes in two parameters: (1) A list of strings called str\_list. (2) An integer value t. The function returns a list that contains the first n strings in str\_list such that ***the total number of digits*** in these n strings is **strictly** larger than t. The function should find the smallest n that satisfies the condition above and return those n strings. If the total number of digits in all the strings in str\_list is still not larger than t, then the function returns a list that contains all the strings in str\_list.

For example, suppose str\_list is ['ab12', 'IS111', '9', 'X7Z', 'k', 'lmn'], and t is 5, then the function should return ['ab12', 'IS111', '9'], because the number of digits in these three strings is 6, which is larger than 5.

**Hint:** It helps to define a function that counts how many digits there are inside a string.