18.1 LAB: Checker for integer string

Forms often allow a user to enter an integer. Write a program that takes in a string representing an integer as input, and outputs yes if every character is a digit 0-9.

Ex: If the input is:


```
1995
```

the output is:

```
yes
```

Ex: If the input is:

```
42,000
```

or any string with a non-integer character, the output is:

```
no
```


LAB ACTIVITY | 18.1.1: LAB: Checker for integer string | 0 / 10

main.py — Load default template...

```
user_string = input()

''' Type your code here. '''

```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.py (Your program) → 0

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.2 LAB: Name format

Many documents use a specific format for a person's name. Write a program whose input is:

firstName middleName lastName

and whose output is:

lastName, firstInitial.middleInitial.

Ex: If the input is:

```
Pat Silly Doe
```


the output is:

```
Doe, P.S.
```

If the input has the form:

firstName lastName

the output is:

lastName, firstInitial.

Ex: If the input is:

```
Julia Clark
```

the output is:


```
Clark, J.
```


LAB ACTIVITY | 18.2.1: LAB: Name format | 0 / 10

main.py — Load default template...

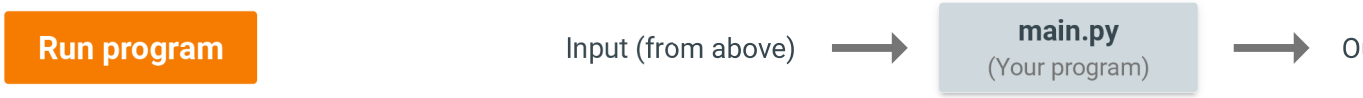
```
1 ''' Type your code here. '''
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.



Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.3 LAB: Count characters

Write a program whose input is a string which contains a character and a phrase, and whose output indicates the number of times the character appears in the phrase. The output should include the input character and use the plural form, n's, if the number of times the characters appears is not exactly 1.

Ex: If the input is:

```
n Monday
```

the output is:

```
1 n
```

Ex: If the input is:

```
z Today is Monday
```

the output is:


```
0 z's
```

Ex: If the input is:

```
n It's a sunny day
```

the output is:

```
2 n's
```

Case matters. n is different than N.

Ex: If the input is:

```
n Nobody
```


the output is:

```
0 n's
```


LAB ACTIVITY | 18.3.1: LAB: Count characters | 0 / 10

main.py — Load default template...

```
1 ''' Type your code here. '''
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.py (Your program) → O

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.4 LAB: Mad Lib - loops

This section's content is not available for print.

18.5 LAB: Palindrome

A palindrome is a word or a phrase that is the same when read both forward and backward. Examples are: "bob," "sees," or "never odd or even" (ignoring spaces). Write a program whose input is a word or phrase, and that outputs whether the input is a palindrome.

Ex: If the input is:

```
bob
```

the output is:

```
bob is a palindrome
```


Ex: If the input is:

```
bobby
```

the output is:

```
bobby is not a palindrome
```

Hint: Start by removing spaces. Then check if a string is equivalent to its reverse.

NaN.2579306.qx3zqy7

LAB ACTIVITY | 18.5.1: LAB: Palindrome | 0 / 10

main.py — Load default template...

```
1 ''' Type your code here. '''
2
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.py** (Your program) → 0

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.6 LAB: Acronyms

An acronym is a word formed from the initial letters of words in a set phrase. Write a program whose input is a phrase and whose output is an acronym of the input. Append a period (.) after each letter in the acronym. If a word begins with a lower case letter, don't include that letter in the acronym. Assume the input has at least one upper case letter.

Ex: If the input is:

```
Institute of Electrical and Electronics Engineers
```

the output is:

```
I.E.E.E.
```

Hint: Use isupper() to check if a letter is upper case.

LAB ACTIVITY | 18.6.1: LAB: Acronyms 0 / 10

main.py Load default template...

```
1 ''' Type your code here. '''
```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.py (Your program) → 0

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.7 LAB: Contains the character

Write a program that reads a character, then reads in a list of words. The output of the program is every word in the list that contains the character at least once. Assume at least one word in the list will contain the given character.

Ex: If the input is:

```
z
hello zoo sleep drizzle
```

the output is:


```
zoo
drizzle
```

Keep in mind that the character 'a' is not equal to the character 'A'.

LAB ACTIVITY | 18.7.1: LAB: Contains the character | 0 / 10

main.py | Load default template...

```
1 ''' Type your code here. '''
```


Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.py (Your program) → 0

Program output displayed here

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

18.8 LAB: Warm up: Parsing strings

(1) Prompt the user for a string that contains two strings separated by a comma. (1 pt)

- Examples of strings that can be accepted:
  - Jill, Allen
  - Jill , Allen
  - Jill,Allen

Ex:

```
Enter input string:
Jill, Allen
```

(2) Report an error if the input string does not contain a comma. Continue to prompt until a valid string is entered. *Note: If the input contains a comma, then assume that the input also contains two strings.* (2 pts)

Ex:

```
Enter input string:
Jill Allen
Error: No comma in string.

Enter input string: Jill, Allen
```

(3) Using string splitting, extract the two words from the input string and then remove any spaces. Output the two words. (2 pts)

Ex:

```
Enter input string:
Jill, Allen
First word: Jill
Second word: Allen
```


(4) Using a loop, extend the program to handle multiple lines of input. Continue until the user enters q to quit. (2 pts)

Ex:

```
Enter input string:
Jill, Allen
First word: Jill
Second word: Allen

Enter input string:
Golden , Monkey
First word: Golden
Second word: Monkey

Enter input string:
Washington,DC
First word: Washington
Second word: DC

Enter input string:
q
```


LAB ACTIVITY | 18.8.1: LAB: Warm up: Parsing strings | 0 / 7

main.py — Load default template...

```
1 # Type your code here
```


Develop mode | Submit mode

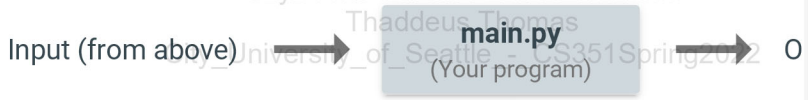
Run your program as often as you'd like, before submitting for grading. Below, type any needed input

values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program



Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

18.9 LAB*: Program: Data visualization

(1) Prompt the user for a title for data. Output the title. (1 pt)

Ex:

```
Enter a title for the data:
Number of Novels Authored
You entered: Number of Novels Authored
```

(2) Prompt the user for the headers of two columns of a table. Output the column headers. (1 pt)

Ex:


```
Enter the column 1 header:
Author name
You entered: Author name

Enter the column 2 header:
```

```
Number of novels
You entered: Number of novels
```

(3) Prompt the user for data points. Data points must be in this format: *string, int*. Store the information before the comma into a string variable and the information after the comma into an integer. The user will enter **-1** when they have finished entering data points. Output the data points. Store the string components of the data points in a list of strings. Store the integer components of the data points in a list of integers. (4 pts)

Ex:

```
Enter a data point (-1 to stop input):
Jane Austen, 6
Data string: Jane Austen
Data integer: 6
```

(4) Perform error checking for the data point entries. If any of the following errors occurs, output the appropriate error message and prompt again for a valid data point.

- If entry has no comma
  - Output: **`Error: No comma in string.`** (1 pt)
- If entry has more than one comma
  - Output: **`Error: Too many commas in input.`** (1 pt)
- If entry after the comma is not an integer
  - Output: **`Error: Comma not followed by an integer.`** (2 pts)

Ex:

```
Enter a data point (-1 to stop input):
Ernest Hemingway 9
Error: No comma in string.

Enter a data point (-1 to stop input):
Ernest, Hemingway, 9
Error: Too many commas in input.

Enter a data point (-1 to stop input):
Ernest Hemingway, nine
Error: Comma not followed by an integer.
```

```
Enter a data point (-1 to stop input):
Ernest Hemingway, 9
Data string: Ernest Hemingway
Data integer: 9
```

(5) Output the information in a formatted table. The title is right justified with a minimum field width value of 33. Column 1 has a minimum field width value of 20. Column 2 has a minimum field width value of 23. (3 pts)

Ex:

```
        Number of Novels Authored
Author name          |       Number of novels
-----------------------------------------------
Jane Austen          |                        6
Charles Dickens      |                       20
Ernest Hemingway     |                        9
Jack Kerouac         |                       22
F. Scott Fitzgerald  |                        8
Mary Shelley         |                        7
Charlotte Bronte     |                        5
Mark Twain           |                       11
Agatha Christie      |                       73
Ian Flemming         |                       14
Stephen King         |                       54
Oscar Wilde          |                        1
```

(6) Output the information as a formatted histogram. Each name is right justified with a minimum field width value of 20. (4 pts)

Ex:

```
         Jane Austen ******
     Charles Dickens ********************
    Ernest Hemingway *********
        Jack Kerouac **********************
 F. Scott Fitzgerald ********
        Mary Shelley *******
    Charlotte Bronte *****
          Mark Twain ***********
     Agatha Christie
```

```
**********************************************************************
        Ian Flemming **************
        Stephen King
**************************************************
         Oscar Wilde *
```


LAB ACTIVITY | 18.9.1: LAB*: Program: Data visualization | 0 / 17

main.py — Load default template...

```
1 # Type your code here
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.py** (Your program) → 0

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```


18.10 LAB: Remove all non-alpha characters

This section's content is not available for print.