UMKC School of Computing and Engineering

Lab Week 5 – Strings

Linda Hall Library

Skills Needed to complete this Lab

- Index, slice strings
- Iterate over strings
- use string methods

Check Digit

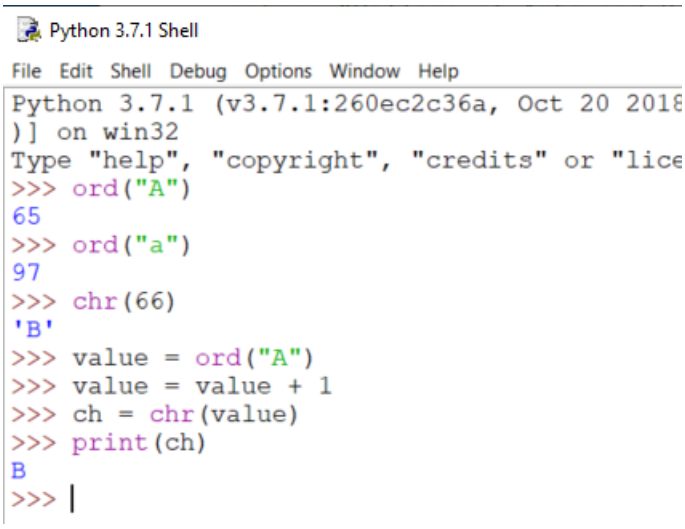
A check digit is a digit that is used in a string or value that determines if the rest of the value is correct. A formula applied to the rest of the values is applied and the value is compared against the check digit, if they are equal, then the number was likely valid. If they are not, then a character may have been entered incorrectly. Check digits are used in ISBN library book numbers, VIN numbers on your car, Credit Card Numbers, and almost any place where data entry errors are likely to happen and must be avoided.

You will want to use functions to break the program down as much as possible into smaller tasks. While you are testing you may also want to show code words so that it is easier to test. We've again given you a set of unit tests to pass. We have not defined the functions in the solution but will give you the function signatures that are required. You may add extra functions.

You should be able to iterate over strings by index, and get a value in a string by index. You also need to be able to convert a string value to a number. All characters in a computer have a number associated with them. For instance, the string character A has a value of 65, while a lowercase a has a value of 97. You can get the integer value of a string character with the ord function. You can convert it back with the chr function as shown below.

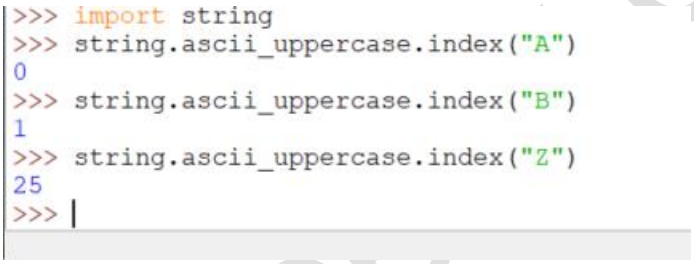
Notice in the sample below, we can get the value of A, then we get the value of a, then we get the character associated with the integer 66. The last part gets the integer value of the character for the upper case A, we add 1 to it, convert it back to a character with chr, and then output the result. The character that is one higher than A is B.

UMKC School of Computing and Engineering



In our program, we need to associate a character with a numeric value to use in calculating our check digit. In our solution an A is worth 0, a B is worth 1, … and a Z is 25. (We will only be using A-Z and 0-9. If ord returns 65 for an A, then how would we convert it to be a 0?

Another way to get a value for a character is to import the string module. The string module has an attribute with all the upper case letters. Since it is a string, you can use the index or find to get the index of a particular character. View the sample below



Either of these would work well in our final solution.

Check Digit Program

The Linda Hall library wants to come up with a new library card numbering system for students. The card number's first 5 characters are A-Z, which will normally be the first five characters of the student's name. The next character at index 5 is a string value of either 1, 2, or 3 which represents the different schools; SCE, School of Law, or College of Arts and Sciences. The character at index 6 is either 1, 2, 3, or 4. These are the grade levels; Freshman, Sophomore, Junior, and Senior. The next 2 characters are 0-9, and the last character at index 9 is the check digit to verify the previous values. The last character is also 0-9.

The formula is the sum of all the values at indexes from 0-8. The value at each position is the value of the character multiplied by the index+1. We take the modulus of the final value and 10 to get the check digit value. Below is an example with the string ABCDE12345

index	0	1	2	3	4	5	6	7	8	9
example	A	B	C	D	E	1	2	3	4	5
value	0	1	2	3	4	1	2	3	4	5
(index+1)*value	1*0	2*1	3*2	4*3	5*4	6*1	7*2	8*3	9*4	-

$1 * 0 + 2 * 1 + 3 * 2 + 4 * 3 + 5 * 4 + 6 * 1 + 7 * 2 + 8 * 3 + 9 * 4 = 120120\%10 = 0$The check digit should be 0. Since this check digit doesn't match the 5 in index 9 it would be an invalid number.

Validations

When a user enters a number, the final item evaluated is the check digit, but we can validate several things before the check digit.

- Length of the card # value is 10 characters
- First 5 characters can only be A-Z
- The 6th character (at index 5) can only be 1, 2 or 3.
- The 7th character (at index 6) can only be 1, 2, 3, or 4.
- The 8th, 9th, and 10th characters (index 7, 8, and 9) can only be [0-9]

Functions to Implement

The following functions must be present as defined and used in the program as intended.

get_school function should take a string as a parameter and return a string based on what school the library card is used for.

Index 5 value	**Returns**
1	School of Computing and Engineering SCE
2	School of Law
3	College of Arts and Sciences
Any other value	Invalid School

get_grade function should take the library card as a string parameter and return a string indicating what grade of the student.

Index 6 Value	**Returns**
1	Freshman
2	Sophomore
3	Junior
4	Senior
5	Invalid Grade

UMKC School of Computing and Engineering

character_value function takes a single character as a string parameter and returns an integer representation of that value. A returns 0, B returns 1, Z returns 25.

get_check_digit function should take the library card as a string parameter and return an integer with the value of the check digit. Using the formula and method above, here are some values for input and responses.

```
>>> get_check_digit("ABCDE1234X")
0
>>> get_check_digit("ZZZZZ1259X")
6
>>> get_check_digit("VWXYZ3459X")
2
>>>
```

verify_check_digit function takes a library card number as a string and returns a tuple (2 values) a boolean and a if the library card is valid then it

Error	Returns
Input string is not length 10	False, "The length of the number given must be 10"
One of the first 5 characters is not [A-Z]. A character % in index 3 is shown.	False, "The first 5 characters must be A-Z, the invalid character is at 3 is %"
One of the characters from idx 7-9 is not [0-9]. An X in index 7 is shown	False, "The last 3 characters must be 0-9, the invalid character is at 7 is X"
Character at index 5 is not a 1, 2 or 3	False "The sixth character must be 1 2 or 3"
Character at index 6 is not a 1, 2, 3, or 4	False, "The seventh character must be 1 2 3 or 4"
The check digit at index 9 is not correct. Example given is check digit 8 in string does not equal the calculated check digit of 2.	False, "Check Digit 8 does not match calculated value 2."
The library card is valid and check digit matches	True, ""

```
>>> verify_check_digit("V2XYZ58BB")
(False, 'The length of the number given must be 10')
>>> verify_check_digit("V2XYZ58BBB")
(False, 'The first 5 characters must be A-Z, the invalid character is at 1 is 2')
>>> verify_check_digit("VWXYZ58BBB")
(False, 'The last 3 characters must be 0-9, the invalid character is at 7 is B')
>>> verify_check_digit("VWXYZ58593")
(False, 'The sixth character must be 1 2 or 3')
>>> verify_check_digit("VWXYZ38593")
(False, 'The seventh character must be 1 2 3 or 4')
>>> verify_check_digit("VWXYZ34593")
(False, 'Check Digit 3 does not match calculated value 2.')
>>> verify_check_digit("VWXYZ34592")
(True, '')
>>>
```

Ln: 38 Col: 4

UMKC School of Computing and Engineering

Unit Tests

You can run the unit tests against your work to check the functionality. It may not mean it's completely correct but will help you in finding common errors. The unit test file is a menu, so you can test each function that you write separately. Each passing test is worth 1 point a piece. The functions character_value, get_school, and get_grade should be the easiest, to begin with.

```
                CHOOSE TEST TO RUN
1. Test function character_value
2. Test function check_digit
3. Test function verify_check_digit
4. Test function get_school
5. Test function get_grade
A. Test All
Q. Quit

==> 1
...
----------------------------------------------------------------------
Ran 3 tests in 0.004s

OK

                CHOOSE TEST TO RUN
1. Test function character_value
2. Test function check_digit
3. Test function verify_check_digit
4. Test function get_school
5. Test function get_grade
A. Test All
Q. Quit

==> a
.............................
----------------------------------------------------------------------
Ran 31 tests in 0.045s

OK
```

Example Program

Once your functions are complete, complete the program under the if __name__ == "__main__" section. Example given below.

```
                        Linda Hall
                    Library Card Check
============================================================

Enter Libary Card.  Hit Enter to Exit ==> XY292
Libary card is invalid.
The length of the number given must be 10

Enter Libary Card.  Hit Enter to Exit ==> 1111111111
Libary card is invalid.
The first 5 characters must be A-Z, the invalid character is at 0 is 1

Enter Libary Card.  Hit Enter to Exit ==> VWXYZ34592
Library card is valid.
The card belongs to a student in College of Arts and Sciences
The card belongs to a Senior

Enter Libary Card.  Hit Enter to Exit ==> BINGH14592
Libary card is invalid.
Check Digit 2 does not match calculated value 0.

Enter Libary Card.  Hit Enter to Exit ==> BINGH14590
Library card is valid.
The card belongs to a student in School of Computing and Engineering SCE
The card belongs to a Senior

Enter Libary Card.  Hit Enter to Exit ==>
```

Grading and Turning In

Turn in your program before the end of the lab. Only upload the lab06.py file, as other files will be ignored.