# Laboratory 10 — User Input, While Loops, Dictionaries

Topics covered:

* User Input
* While Loops
* Dictionaries

## preparation

Lab attendance is compulsory. You will receive 1 mark for being present at the start of the lab and staying at least until the tutor has finished introducing the lab and has signed your attendance sheet.

## Exercises

The following exercises must be completed during your allocated laboratory time. You must show your work to the laboratory tutor who will sign off when the work is completed correctly. Include all exercises in the single module (file) named Lab10.py which you can find in the Lab10 folder.

You are given the following functions in the Lab10.py file.

* The spy() function which takes the string that you wish to encode as a parameter. It prints out the string to be encoded. Then it deciphers the code and prints it again. It calls the master\_encode() and master\_decode() functions.
* The master\_encode() function which takes a string as the only parameter and returns both the string encoded in the cipher, and also the master\_code encoded in the cipher. It calls the shuffle(), make\_cipher(), and encode() functions.
* Themaster\_decode() function which takes two strings as parameters, the encoded basestring and the encoded string. It returns the second parameter (the encoded string) after it has been decoded. It calls the functions histogram() and decode().
* The function encode() which takes a string and a shuffled alphabet as parameters and returns the string encoded with that shuffled alphabet.

### Exercise 10.1

[3 marks] Write a function named decode() which takes a string and two dictionaries as parameters and returns the decoded string. It decodes the string by looping through the letters in the parameter string, and for each letter, it extracts the number value for the letter from the second dictionary, then loops through the first dictionary until it finds a letter whose number value matches this number value. It then concatenates this letter on to the string to be returned.

**Arguments:** a string and two dictionaries

**Returns:** decoded string

**Sample Tests:**

|  |
| --- |
| >>> decode("twlwjmzjdkjzplmva", {'t': 20, 'u': 21, 'v': 22, 'w': 23, 'p': 16, 'q': 17, 'r': 18, 's': 19, 'x': 24, 'y': 25, 'z': 26, 'd': 4, 'e': 5, 'f': 6, 'g': 7, ' ': 27, 'a': 1, 'b': 2, 'c': 3, 'l': 12, 'm': 13, 'n': 14, 'o': 15, 'h': 8, 'i': 9, 'j': 10, 'k': 11}, {'t': 8, 'u': 3, 'v': 14, 'w': 5, 'p': 20, 'q': 11, 'r': 26, 's': 22, 'x': 23, 'y': 24, 'z': 19, 'd': 13, 'e': 6, 'f': 2, 'g': 10, ' ': 21, 'a': 7, 'b': 17, 'c': 15, 'l': 18, 'm': 9, 'n': 4, 'o': 1, 'h': 16, 'i': 12, 'j': 27, 'k': 25})  'here is my string'  >>> decode("cxnlvbujwlumxejlnjqnkb", {'t': 20, 'u': 21, 'v': 22, 'w': 23, 'p': 16, 'q': 17, 'r': 18, 's': 19, 'x': 24, 'y': 25, 'z': 26, 'd': 4, 'e': 5, 'f': 6, 'g': 7, ' ': 27, 'a': 1, 'b': 2, 'c': 3, 'l': 12, 'm': 13, 'n': 14, 'o': 15, 'h': 8, 'i': 9, 'j': 10, 'k': 11}, {'t': 21, 'u': 18, 'v': 8, 'w': 19, 'p': 11, 'q': 3, 'r': 22, 's': 13, 'x': 14, 'y': 12, 'z': 25, 'd': 16, 'e': 7, 'f': 23, 'g': 24, ' ': 10, 'a': 26, 'b': 5, 'c': 1, 'l': 20, 'm': 9, 'n': 15, 'o': 6, 'h': 17, 'i': 2, 'j': 27, 'k': 4})  'another string to code' |

### Exercise 10.2

[2 marks] Write a function named histogram() which takes a string as the only parameter and returns a dictionary that represents a histogram of letter counts.

**Arguments:** a string

**Returns:** returns a histogram dictionary of letter counts

**Sample Test:**

|  |
| --- |
| >>> histogram("abbcccddddeeeeeffffffggggggghhhhhhhhiiiiiiiiijjjjjjjjjjkkkkkkkkkkkllllllllllllmmmmmmmmmmmmmnnnnnnnnnnnnnnoooooooooooooooppppppppppppppppqqqqqqqqqqqqqqqqqrrrrrrrrrrrrrrrrrrsssssssssssssssssssttttttttttttttttttttuuuuuuuuuuuuuuuuuuuuuvvvvvvvvvvvvvvvvvvvvvvwwwwwwwwwwwwwwwwwwwwwwwxxxxxxxxxxxxxxxxxxxxxxxxyyyyyyyyyyyyyyyyyyyyyyyyyzzzzzzzzzzzzzzzzzzzzzzzzzz")  {'f': 6, 'g': 7, 'd': 4, 'e': 5, 'b': 2, 'c': 3, ' ': 27, 'a': 1, 'n': 14, 'o': 15, 'l': 12, 'm': 13, 'j': 10, 'k': 11, 'h': 8, 'i': 9, 'v': 22, 'w': 23, 't': 20, 'u': 21, 'r': 18, 's': 19, 'p': 16, 'q': 17, 'z': 26, 'x': 24, 'y': 25}  >>> histogram("ijngwggjnmmxqhlr cxhkiqonnswuurnisljstzhkcbonbsybwrtmsmkoab sjqiojohzjiksnpiqpwsslifkblannntspgniikjgaiwkjnocxoayh khmrghwxdbkiqlglrh khcxtpgjhnjsybmobkkbjoqpafujqhqolbmoho uuhbnxkhqg snxiswoubqsxhhztksbhdonttrynmlskoi so ultoblun qnebholidivjh lgjqmjrduhqtigigkjmtkc bxbymaqbwouxlihiqkqlmghnwton hwsotcrfihisqolibqroqlnlnlmcwtjbgu fxmwtvrowqylygludqaknqlycngilihmsb ghmcpo jool")  {'f': 4, 'g': 17, 'd': 5, 'e': 1, 'b': 21, 'c': 9, ' ': 15, 'a': 7, 'n': 25, 'o': 27, 'l': 23, 'm': 16, 'j': 18, 'k': 19, 'h': 26, 'i': 24, 'v': 2, 'w': 13, 't': 14, 'u': 12, 'r': 10, 's': 20, 'p': 6, 'q': 22, 'z': 3, 'x': 11, 'y': 8}  41 |

### Exercise 10.3

[3 marks] Write the shuffle() function which takes a string parameter and returns the string shuffled in a random order.

**Arguments**: the string to be shuffled

**Returns:** a randomly shuffled string

**Sample Tests:**

|  |
| --- |
| >>> shuffle('abbcccddddeeeeeffffffggggggghhhhhhhhiiiiiiiiijjjjjjjjjjkkkkkkkkkkkllllllllllllmmmmmmmmmmmmmnnnnnnnnnnnnnnoooooooooooooooppppppppppppppppqqqqqqqqqqqqqqqqqrrrrrrrrrrrrrrrrrrsssssssssssssssssssttttttttttttttttttttuuuuuuuuuuuuuuuuuuuuuvvvvvvvvvvvvvvvvvvvvvvwwwwwwwwwwwwwwwwwwwwwwwxxxxxxxxxxxxxxxxxxxxxxxxyyyyyyyyyyyyyyyyyyyyyyyyyzzzzzzzzzzzzzzzzzzzzzzzzzz')  'jvvlwmvtvxynktytpytiqitttotxvtgwompkqehrpfxzuzuxxiuzwvkuxnuzzkspynqvcszwyeogsqijyskoyxuvzlvzsynykzwsyqsktxwvloywqvyorswruulznwqmwnovsxxfwcuqtzhqnrxlvnouenxusyqhokmyxpzshtjfuvnwbczjppzyqjrqyexsfrpqyuuxjrmhzhtzsvswuqrsltkdkflqsmlwkpgplarvuzjxbdzxjvtsymqyxrqwoeomwyunzirzphzitzozmdtxgwrmlvjjrxsznptgwuoywprmyvpopohvgrgytxswwxvwmlnimluztnvrxwdpuxriuizyfry'  >>> shuffle('abcdefghijklmnopqrstuvwxyz ')  'kjxyq virpadmnlcozsugwthfbe' |

### Exercise 10.4

[1 mark] Run the spy() function with a message of your choice to show that your functions decode(), histogram(),shuffle()work correctly with the other functions.

## Homework Exercises

### The following exercises must be completed prior to the start of your Lab 11 session. Include all the exercises in the single module (file), named "Lab10\_Homework.py" which you can find in the Lab10 folder. Your file must include a docstring at the top of the file containing your name, UPI and ID number. You must submit the file containing your exercises using the Assignment Dropbox before the start of Lab 11.

### Exercise 10.5

[5 marks] Write the play\_sudoku() function which allows you to enter numbers into the sudoku puzzle. This should work for a 9x9 board. The function should redraw the board after each entry and warn you if it determines that you have made an illegal move or that it does not think your entry is a forced move. The functions in the provided Lab10\_Homework.py filewill be of help. You only have to write the top level function play\_sudoku(). An “illegal move” is one where you enter a number which is already present in the row, column or square; this is tested for using the provided function test\_sudoku(). A “forced move” is when there is only one possible number which can be played in the combined row, column or square; this is tested for using the provided function calculate().

**Arguments:** a list containing the sudoku puzzle to solve.

**Prints:** a list representing the new board after each attempt.

**Other knowledge**: You can use functions from Lecture 24, which you can find in the provided Lab10\_Homework.py file.

**Sample Tests:**

|  |
| --- |
| >>> play\_sudoku([0, 9, 8, 2, 7, 0, 0, 3, 0, 3, 0, 4, 0, 0, 6, 0, 0, 7, 1, 0, 0, 5, 0, 0, 2, 0, 4, 0, 0, 5, 0, 0, 0, 6, 0, 1, 7, 1, 0, 0, 0, 0, 0, 8, 9, 4, 0,3, 0, 0, 0, 5, 0, 0, 6, 0, 1, 0, 0, 8, 0, 0, 3, 2, 0, 0, 1, 0, 0, 4, 0, 8, 0, 3,0, 0, 4, 2, 7, 1, 0])  0 9 8 2 7 0 0 3 0  3 0 4 0 0 6 0 0 7  1 0 0 5 0 0 2 0 4  0 0 5 0 0 0 6 0 1  7 1 0 0 0 0 0 8 9  4 0 3 0 0 0 5 0 0  6 0 1 0 0 8 0 0 3  2 0 0 1 0 0 4 0 8  0 3 0 0 4 2 7 1 0  Enter number or q to quit: 5  Enter row (0-8): 0  Enter col (0-8): 0  Good choice  5 9 8 2 7 0 0 3 0  3 0 4 0 0 6 0 0 7  1 0 0 5 0 0 2 0 4  0 0 5 0 0 0 6 0 1  7 1 0 0 0 0 0 8 9  4 0 3 0 0 0 5 0 0  6 0 1 0 0 8 0 0 3  2 0 0 1 0 0 4 0 8  0 3 0 0 4 2 7 1 0  Enter number or q to quit: 5  Enter row (0-8): 0  Enter col (0-8): 1  Not a legal location. Cell already contains a number.  5 9 8 2 7 0 0 3 0  3 0 4 0 0 6 0 0 7  1 0 0 5 0 0 2 0 4  0 0 5 0 0 0 6 0 1  7 1 0 0 0 0 0 8 9  4 0 3 0 0 0 5 0 0  6 0 1 0 0 8 0 0 3  2 0 0 1 0 0 4 0 8  0 3 0 0 4 2 7 1 0  Enter number or q to quit: 5  Enter row (0-8): 0  Enter col (0-8): 5  Not a forced move  Should I do it anyway? (enter y or n): y  You have made a mistake  Do you want to return to original board or quit (enter y or q): y  0 9 8 2 7 0 0 3 0  3 0 4 0 0 6 0 0 7  1 0 0 5 0 0 2 0 4  0 0 5 0 0 0 6 0 1  7 1 0 0 0 0 0 8 9  4 0 3 0 0 0 5 0 0  6 0 1 0 0 8 0 0 3  2 0 0 1 0 0 4 0 8  0 3 0 0 4 2 7 1 0  Enter number or q to quit: q  >>> play\_sudoku([0, 9, 8, 2, 7, 4, 1, 3, 6, 3, 2, 4, 9, 1, 6, 8, 5, 7, 1, 7, 6, 5, 8, 3, 2, 9, 4, 9, 8, 5, 3, 2, 7, 6, 4, 1, 7, 1, 2, 4, 6, 5, 3, 8, 9, 4, 6, 3, 8, 9, 1, 5, 7, 2, 6, 4, 1, 7, 5, 8, 9, 2, 3, 2, 5, 7, 1, 3, 9, 4, 6, 8, 8, 3, 9, 6, 4, 2, 7, 1, 5])  0 9 8 2 7 4 1 3 6  3 2 4 9 1 6 8 5 7  1 7 6 5 8 3 2 9 4  9 8 5 3 2 7 6 4 1  7 1 2 4 6 5 3 8 9  4 6 3 8 9 1 5 7 2  6 4 1 7 5 8 9 2 3  2 5 7 1 3 9 4 6 8  8 3 9 6 4 2 7 1 5  Enter number to enter or (q) to quit: 5  Enter row (0-8): 0  Enter col (0-8): 0  Good choice  You solved it!  >>> |

### Exercise 10.6

[5 marks] Extend the play\_rock\_paper() code given in Lecture 24 Slide 8 so that it functions as in the Sample Test below. You need to store (in a histogram dictionary) all pairs of plays the user has made. For instance, if there are 4 times that the user has played “paper” after “rock” the dictionary would contain {'rp' : 4}. This is basically how the player <http://www.nytimes.com/interactive/science/rock-paper-scissors.html?_r=0> works. You have just written a program that does “machine learning”. You are an Artificial Intelligence Programer. Congratulations!

In the Lab10\_Homework.py file you will be given the functions: who\_wins(), find\_best\_play() and find\_mapping().

You need to write the top\_level function play\_rock\_paper() which will be an extension of the one given in class.

So for instance in the example below once it determines that after playing “rock” you will play “paper”, and after playing “paper” you will play “scissors” and after playing “scissors” you will play “rock”, it learns to anticipate your next move from your last move. Basically, in this instance it is because the dictionary entry for "rp" is larger than the dictionary entry for "rr" or "rs". As you can see, after the first few moves, the computer plays fairly well. It has learned the pattern.

**Arguments:** None

**Returns:** Nothing

**Sample Test:**

|  |
| --- |
| >>> play\_rock\_paper()  Choose "r" "p" or "s": r  Computer chose paper.  You chose rock.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": p  Computer chose rock.  You chose paper.  You win.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": s  Computer chose rock.  You chose scissors.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": r  Computer chose rock.  You chose rock.  We are tied.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": p  Computer chose scissors.  You chose paper.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": s  Computer chose rock.  You chose scissors.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": r  Computer chose paper.  You chose rock.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": p  Computer chose scissors.  You chose paper.  Computer wins.  Do you want to play again? Answer "y" or "n": y  Choose "r" "p" or "s": s  Computer chose rock.  You chose scissors.  Computer wins.  Do you want to play again? Answer "y" or "n": n  >>> |

## Advanced Exercises (optional)

### Exercise 10.7

[0 marks] Write a function named reverse\_word\_game() which takes no parameters. It is similar to the code on silde 7 in lecture 25. But now you must play a word whose “last letter” is the same as the first letter your opponent just played. This will be very easy for the computer, but very hard for you!!

**Sample test:**

|  |
| --- |
| >>> reverse\_word\_game()  Pick the first word: cat  This is my word: c  Give next word (or q to quit): hic  Your word is illegal  Give next word (or q to quit): sac  This is my word: s  Give next word (or q to quit): his  This is my word: h  Give next word (or q to quit): length  This is my word: l  Give next word (or q to quit): yell  This is my word: y  Give next word (or q to quit): spy  This is my word: s's  Give next word (or q to quit): tis  Your word is illegal  Give next word (or q to quit): sis  This is my word: sa  Give next word (or q to quit): less  This is my word: l'oeil  Give next word (or q to quit): q  >>> |

### Exercise 10.8

[0 marks] Write a function named black\_jack(). It is fairly easy to write a function that plays blackjack especially if you do not use split hands when you get two of the same cards (which my code does not). You can find out how to play Blackjack at <http://en.wikipedia.org/wiki/Blackjack>

**Sample test:**

|  |
| --- |
| >>> blackjack()  This is your hand ['13D', '6S']  Do you want another card? Answer "y" or "n": y  This is your hand now ['13D', '6S', '10S']  Do you want another card? Answer "y" or "n": n  Your hand ['13D', '6S', '10S']  Computer hand ['9D', '5H', '7D']  Computer wins  Do you want to play again? Answer "y" or "n": y  This is your hand ['13C', '1S']  Do you want another card? Answer "y" or "n": y  This is your hand now ['13C', '1S', '11S']  Do you want another card? Answer "y" or "n": n  Your hand ['13C', '1S', '11S']  Computer hand ['12C', '13H']  Computer wins  Do you want to play again? Answer "y" or "n": y  This is your hand ['9C', '13H']  Do you want another card? Answer "y" or "n": n  Your hand ['9C', '13H']  Computer hand ['5D', '8S', '9H']  You win  Do you want to play again? Answer "y" or "n": n  >>> |

### Exercise 10.9

[0 marks] Write a function named play\_lingo() which plays a mastermind type game with letters.

A description of the game can be found at <http://www.ling.gu.se/~lager/python_exercises.html>.

**Sample test:**

|  |
| --- |
| >>> play\_lingo()  I have chosen a word  My word has 3 letters  What is your guess: asd  Here is a clue: as[d]  What is your next guess(or q to quit): four  Your guess is not the right length.  What is your next guess(or q to quit): wed  Here is a clue:we[d]  what is your next guess(or q to quit): rid  here is a clue: ri[d]  what is your next guess(or q to quit): hud  here is a clue: hu[d]  what is your next guess(or q to quit): pod  here is a clue: p[o][d]  What is your next guess: (or q to quit): mod  Here is a clue: m[o][d]  What is your next guess: (or q to quit): nbv  Here is a clue: nbv  what is your next guess: (or q to quit): cxz  Here is a clue: cxz  What is your next guess: (or q to quit): lkj  Here is a clue: lkj  What is your next guess: (or q to quit): hgf  Here is a clue: h(g)f  What is your next guess: (or q to quit): god  Here is a clue: [g][o][d]  Congratulations  I have chosen a word  My word has 6 letters  What is your guess: q  >>> |

## ASSESSMENT

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab day and time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Check list for laboratory exercises (to be completed by Lab tutor)**

|  |  |
| --- | --- |
| On time: 🞎 (1 mark)  Exercise 10.1: 🞎 (3 marks)  Exercise 10.2: 🞎 (2 marks)  Exercise 10.3: 🞎 (3 marks)  Exercise 10.4: 🞎 (1 mark) | Teaching Assistant: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Total mark: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/10 Tutor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Marking Scheme

|  |  |
| --- | --- |
| Marks | Feedback |
| 0.5 | Includes a docstring at the top of the file containing your name, UPI and ID number. |
| 0.5 | Uses good descriptive variable names. |
| 1 | Include all the exercises in the single file named Lab10 \_Homework.py |
| 1 | The play\_sudoku() function is defined correctly. |
| 3 | Well done! Your play\_sudoku() function passed test cases 1, 2, and 3. |
| 1 | The play\_rock\_paper() function is defined correctly. |
| 3 | Well done! Your play\_rock\_paper() function passed test cases 1, 2 and 3. |