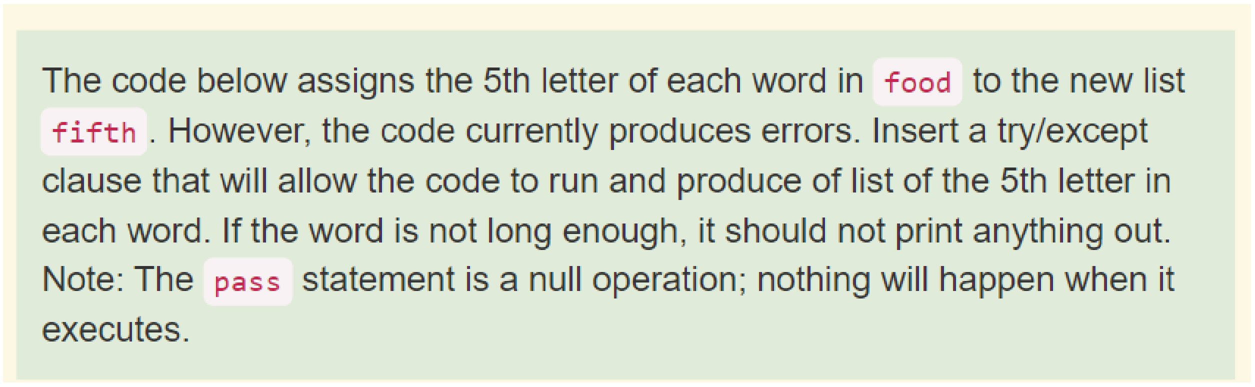
Practice Set-5

No of questions:15

Q1 .Given a faulty program for calculating the maximum number out of a list. Write another program which correctly returns the maximum element from the list. Make 20 iterations and using assertions, check if the output of faulty function equals the output from the correct function. Print correct or incorrect respectively

```
Def faulty_function(inputList):
        Return random.choice(inputList)
```

Q2.

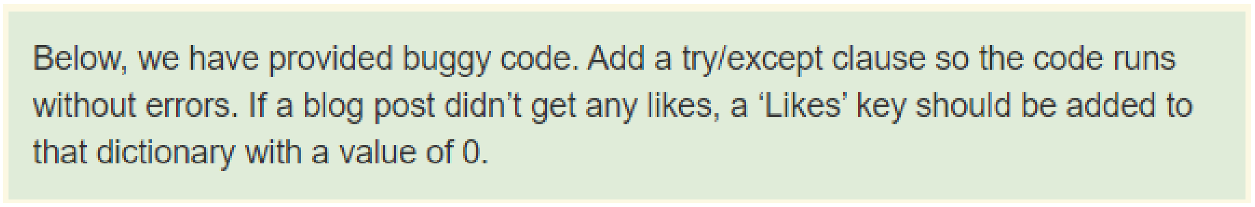


The code below assigns the 5th letter of each word in `food` to the new list `fifth`. However, the code currently produces errors. Insert a try/except clause that will allow the code to run and produce of list of the 5th letter in each word. If the word is not long enough, it should not print anything out. Note: The `pass` statement is a null operation; nothing will happen when it executes.

```
food = ["chocolate", "chicken", "corn", "sandwich", "soup", "potatoes", "beef", "lox", "lemonade"]
fifth = []

for x in food:
   fifth.append(x[4])
```

Q3.



Below, we have provided buggy code. Add a try/except clause so the code runs without errors. If a blog post didn't get any likes, a 'Likes' key should be added to that dictionary with a value of 0.

```
blog_posts = [{'Photos': 3, 'Likes': 21, 'Comments': 2}, {'Likes': 13, 'Comments': 2, 'Shares': 1},
{'Photos': 5, 'Likes': 33, 'Comments': 8, 'Shares': 3}, {'Comments': 4, 'Shares': 2}, {'Photos': 8,
'Comments': 1, 'Shares': 1}, {'Photos': 3, 'Likes': 19, 'Comments': 3}]
```

```
total_likes = 0

for post in blog_posts:
    total_likes = total_likes + post['Likes']
```

Reference: https://runestone.academy/ns/books/published/fopp/Exceptions/Exercises.html

Q4. Design a program for our college, take user input for name and branch. Use try-except and assert for evaluating the correctness of branch (valid branches are CSE ECE CSAM CSD CSSS CSB CSAI). If the input is correct then display a welcome message for them, otherwise display an error message.

Q5. Add try-except to calculate the total percentage of subjects present in both the dictionary. But if a subject is present in marks_obtained but not in total_marks then display an error message.
Example -
Marks Obtained : {"Physics": 82, "Chemistry": 85, "Maths": 92}
Total Marks : {"Physics": 100, "Chemistry": 100, "Maths": 100}

Total Percentage : 86.33

Q6. Design a program that takes input of name(string), rollno(string), mail id(string) and marks(int) for 3 subjects(introduction to programming, Digital Circuits, Maths) check if the input are of mentioned data type if not then it should give an exception else show the details of the student with average marks for the student.

Q7. Provided is a buggy for loop that tries to accumulate some values out of some dictionaries. Insert a try/except so that the code passes.

```
di = [{"Puppies": 17, 'Kittens': 9, "Birds": 23, 'Fish': 90, "Hamsters": 49}, {"Puppies": 23, "Birds": 29, "Fish": 20, "Mice": 20, "Snakes": 7}, {"Fish": 203, "Hamsters": 93, "Snakes": 25, "Kittens": 89}, {"Birds": 20, "Puppies": 90, "Snakes": 21, "Fish": 10, "Kittens": 67}]
total = 0
for diction in di:
    total = total + diction['Puppies']

print("Total number of puppies:", total)
```

Q8. valid number can be split up into these components (in order):

1. A decimal number or an integer.
2. (Optional) An 'e' or 'E', followed by an integer.

A decimal number can be split up into these components (in order):

1. (Optional) A sign character (either '+' or '-').
2. One of the following formats:
    1. One or more digits, followed by a dot '.'.
    2. One or more digits, followed by a dot '.', followed by one or more digits.
    3. A dot '.', followed by one or more digits.

An integer can be split up into these components (in order):

1. (Optional) A sign character (either '+' or '-').
2. One or more digits.

    Example: all the following are valid numbers: ["2", "0089", "-0.1", "+3.14", "4.", "-.9", "2e10", "-90E3", "3e+7", "+6e-1", "53.5e93", "-123.456e789"], while the following are not valid numbers: ["abc", "1a", "1e", "e3", "99e2.5", "--6", "-+3", "95a54e53"].

Given a string s, return true if s is a valid number.

Link- https://leetcode.com/problems/valid-number/

Q9. Given an integer array of even length arr, return true *if it is possible to reorder* arr *such that* arr[2 * i + 1] = 2 * arr[2 * i] *for every* 0 <= i < len(arr) / 2*, or* false *otherwise*.

Link- https://leetcode.com/problems/array-of-doubled-pairs/

Q10. You're going to write an interactive calculator! User input is assumed to be a formula that consist of a number, an operator (at least + and -), and another number, separated by white space (e.g. 1 + 1). Split user input using str.split(), and check whether the resulting list is valid:

- If the input does not consist of 3 elements, raise a FormulaError, which is a custom Exception.
- Try to convert the first and third input to a float (like so: float_value = float(str_value)). Catch any ValueError that occurs, and instead raise a FormulaError
- If the second input is not '+' or '-', again raise a FormulaError

If the input is valid, perform the calculation and print out the result. The user is then prompted to provide new input, and so on, until the user enters quit.

```
>>> 1 + 1
2.0
>>> 3.2 - 1.5
1.7000000000000002
>>> quit
```

Link:https://python.cogsci.nl/basic/exceptions-solution/

Q11.Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Link-https://leetcode.com/problems/valid-parentheses/

Q12. Given a list of strings which may or may not contain integers, return a list of Integers. Use Try/except instead of type checking

Egs: input - [‘hello’, ‘60’, ‘50.5’, ‘17’]

Q13. Given a IPV4 address, check whether its a valid IP address. If it isn’t an error should be thrown. Otherwise, it should return ‘It is Valid’

Egs: 255.1.1.156 is valid IPv4 address but 267.1.356 is not

Q14. valid **MAC address** must satisfy the following conditions:

1. It must contain 12 hexadecimal digits.
2. One way to represent them is to form six pairs of the characters separated with a hyphen (-) or colon(:). For example, 01-23-45-67-89-AB is a valid MAC address.
3. Another way to represent them is to form three groups of four hexadecimal digits separated by dots(.). For example, 0123.4567.89AB is a valid MAC address.

Given a text file ‘macs.txt’ which contains several mac addresses line by line, you have to read the file and check for valid and invalid mac addresses using try except block.

Q15.Create a program which takes in data from the user for registration of Voter ID cards. Name, Age and mobile number of the user will be taken as inputs. Make sure you check for the validity of the data using try except block (For example, phone number has to be a 10 digit integer without any leading zeros).