

BE101-05 INTRODUCTION TO COMPUTING AND PROBLEM SOLVING

Of Dr. APJ Abdul Kalam Technological University (KTU)

Python: Laboratory Question Samples

(Unofficial) Prepared by [Arunanand T A](#) | [Suresh K](#)

As duplicating the old, typical C laboratory problems to the Python laboratory would mean a serious detour from the philosophy of python, the following sample of lab exercises are suggested to incorporate. Please encourage students to make their programs into functions as much as possible, as that's fairly the philosophy of python. And the practice of giving comments in the program may also be promoted. Further, let them write the python code as per the [PEP format](#).

This is an unofficial document, and are suggestions by two python programmers. They are just directives.

Note: Questions might not be in order of the syllabus.

1. Find all numbers which are multiple of 17, but not the multiple of 5, between 2000 and 2500?
2. Swap two integer numbers using a temporary variable. Repeat the exercise using the code format: `a, b = b, a`. Verify your results in both the cases.
3. Given two pairs of Cartesian points such as (x_1, y_1) and (x_2, y_2) . Find the Euclidian distance between them.

Hint: Use `math` module to find the square root.

4. Print the first 2 and last 3 characters in a given string. Use the string slicing concept. Do not use loops. If the length of the string is less than 5, print a suitable message.

5. Implement bubble sort. Do not use the default `sort()` method.

Hint: So as to familiarize with the concept of sorting, and nested looping structures.

6. Implement shallow copy and deep copy of a list. You may use the `copy` module.

Hint: This is a very important concept. While we copy a list, just a reference is copied. Hence if we make any changes to one of the lists, the same will reflect in the other as well. This is called shallow copying. Hence, in some cases we might need to deep copy, where a completely independent copy is created. This can be achieved through the `deepcopy()` method of `copy` module.

7. Write a temperature converter program, which is menu driven. Each such conversion logic should be defined in separate functions. The program should call the respective function based on the user's requirement. The program should run as long as the user wishes so.

8. Find the largest of n numbers, using a user defined function `largest()`

9. Write a function that capitalizes all vowels in a string.

Hint: Do not use the ASCII concept. Use the `upper()` method.

10. Write a function `leapYear()` which receives a four digit year and returns a Boolean value: `True` if the year is leap, `False` if the year is not leap.

11. Read a line containing digits and letters. Write a program to give the count of digits and letters.

Hint: Instead of checking ASCII, use the in-built methods like `isdigit()`, `isalpha()` etc.

12. Write a function `myReverse()` which receives a string as an input and returns the reverse of the string.

13. Use the list comprehension methodology in python, to generate the squares of all odd numbers in a given list.

Hint: [List comprehension](#) is one of the powerful techniques in python; hence it's best if students are exposed to it.

14. Check if a given string is palindrome or not.

Hint: do not use the C philosophy where we compare indices. Instead, copy the string as a new list, reverse the list using `reverse()`, join the list so that the reversed string is formed, using `join()`. Compare the new string and the old one.

15. Write a function to see if a given number is prime or not. Do not use any flag variables. Use `math` module to find the square root, and its roof which will be fed in to `range()`.

Hint: Just the return statements are enough. No need of flag variables. The loop has to run up to the roof of the square root of the number.

16. Write a function to find the factorial of a number using recursion.

17. Extend the above problem to find the nCr of given values of n and r . Verify your result with the help of `itertools` module, which helps find the combinations.
18. Write a program that eliminates duplicates in a list. Do not use the concept of sets. Now, convert the original list into a set. Verify your result in both cases.
19. The user will enter five integers separated from commas. Write a program to read these values, and make a list. Print the list.
- Hint: They will need to read the input using `raw_input()`, and then split the one and only line of input using `split()`. Then each of the values will need to be appended to a list, which will be empty at first.
20. Generate a dictionary and print the same. The keys of the dictionary should be integers between 1 and 10 (both inclusive). The values should be the cubes of the corresponding keys.
21. Create a nested dictionary. The roll number of a student maps to a dictionary. This inner dictionary will have name, age, and place as keys. Read details of at least three students.
- Hint: A sample output should look like the one given below:
- ```
{11: {'name': 'Sachin', 'age': 18, 'place': 'Kochi'}, 12:
{'name': 'Ammu', 'age': 19, 'place': 'Kannur'}, 13:
{'name': 'jishad', 'age': 20, 'place': 'Calicut'}}
```
22. Enter a word. Create a dictionary with the letters of this word as keys, and the corresponding ASCII values as values.
- Hint: Students may use the `ord()` function. Further, this is a simple problem, if they use list comprehension.

23. Define a class with three methods: readString(), printString(), writeString().

The first method should read the contents of a file. The second method should print the contents to the console. The third method should write the contents to a new file.

24. Write a python class to reverse a sentence word by word.

Hint: That is: "I am here" should be reversed as "here am I". The solution is so simple in python. This will help the students understand how powerful python is:

```
class word_by_word_reverse:

 def reverse_words(self, s):

 return ' '.join(reversed(s.split()))
```

```
*** **
```