VINUNIVERSITY

LAB PRACTICE № 7 (SOLUTION): LISTS, ITERATION, AND LOOPS

COMP1010 Introduction to Programming **Week 08**

Lab Practice Submission Instructions:

- This is an individual lab practice and will typically be assigned in the laboratory (computer lab).
- Your program should work correctly on all inputs. If there are any specifications about how the program should be written (or how the output should appear), those specifications should be followed.
- Your code and functions/modules should be appropriately commented. However, try to avoid making your code overly busy (e.g., include a comment on every line).
- Variables and functions should have meaningful names, and code should be organized into functions/methods where appropriate.
- Academic honesty is required in all work you submit to be graded. You should **NOT** copy or share your code with other students to avoid plagiarism issues.
- Use the template provided to prepare your solutions.
- You should upload your .py file(s) to the Canvas **before the end of the laboratory session** unless the instructor gave a specified deadline.
- Submit separate .py file for each Lab problem with the following naming format, for example: **V202000999_Lab7.py**. **Note:** If you are working on Jupiter Notebook, you need to download/convert it to Python .py file for submission.
- Late submission of lab practice without an approved extension will incur the following penalties:

  (a) No submission by the deadline will incur 0.25 point deduction for each problem (most of the problems are due at the end of the lab session).

  (b) The instructor will deduct an additional 0.25 point per problem for each day past the deadline.

  (c) The penalty will be deducted until the maximum possible score for the lab practice reaches zero (0%) unless otherwise specified by the instructor.

Problem 1 – Check prime number

Write a function named `is_prime`, which takes an integer as an argument and returns **true** if the argument is a prime number, or **false** otherwise. Also, write the program that displays prime numbers between 1 to 200. Submit your program to CMS.

Sample output:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107
109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199
```

Problem 2 – Creating an acronym generator

Write a program that creates an acronym from the string entered by the user. Assume that the string only consists of alphabets and spacing. The acronym should be a combination of first letter of each word in the string. However, the following words must be excluded from the acronym: `"but", "do", "not", "to", "has", "have", "had", "then", "who", "when", "is", "why", "what", "how", "while", "hence", "I", "you", "he", "she", "it", "a", "for", "by", "an", "am", "the", "so", "and", "my", "are", "in", "at", "on".`

Exp 1: the acronym for "I do not want to go to bed because I am NOT TIRED" is **WGBBT**
Exp 2: the acronym for "He had excellent skills in playing tennis and badminton" is **ESPTB**

Submit your code to CMS.

Problem 3 – Sum of zero ending numbers

Write the definition of a function `zero_ending(scores)` to add all those values in the list of scores, which are ending with zero and display the sum. For example: If the scores contain `[200, 456, 300, 100, 234, 678]`, the sum should be displayed as 600.

Problem 4 – Nested Loops and Matrices

Given two matrices:

$$\begin{pmatrix} 12 & 7 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

and

$$\begin{pmatrix} 5 & 8 & 1 \\ 6 & 7 & 3 \\ 4 & 5 & 9 \end{pmatrix}$$

Program to add two matrices above using nested loop and submit your solution to CMS. The correct output is formatted as below:

```
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
```

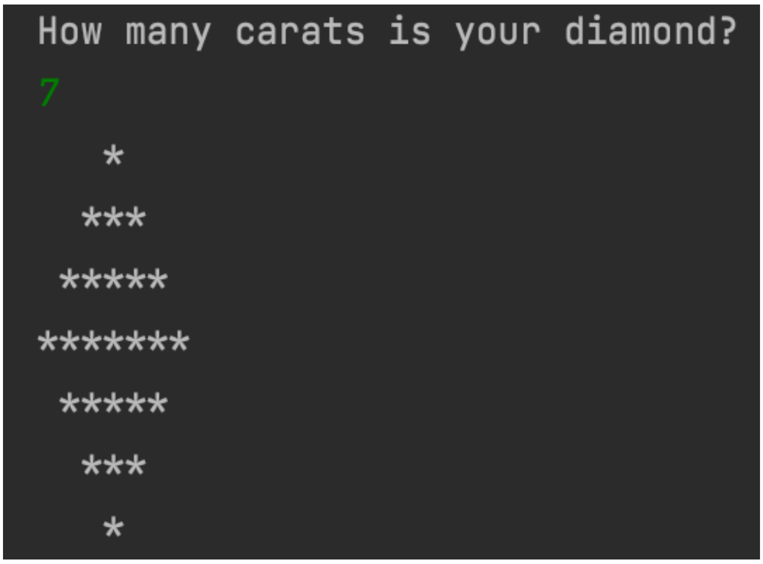
Problem 5 – Make a diamond using nested for loops

[CMS is not required] Write a problem that prints out a diamond. Let's define diamonds for the positive odd integers, where the integer represents the number of rows and the number of stars on tin the middle row. Here are diamonds of size 1, 3, 5 and 7:

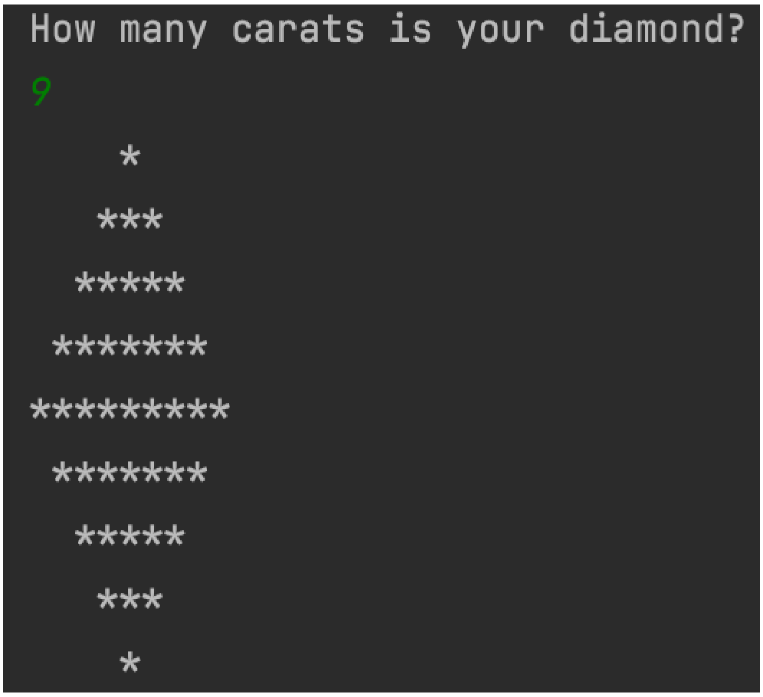
```
   *           *           *           *
              ***         ***         ***
               *         *****       *****
                          ***       *******
                           *         *****
                                      ***
                                       *
```

We notice that the number of spaces decreases by 1 for the first half of the design and increases by 1 in the second half of the design. Secondly, the number of stars increases by 2 in the first half of the design and decreases by 2 in the second half of the design. We'll use our for loop counting variable to represent the number of spaces on each line. In addition, we'll add a minimal bit of error checking in our program. If the user enters an even number, we'll simply tell them that we only make odd sized diamonds. Write this program without using the '*' operator for repeatedly printing out a string.

Sample output:



```
How many carats is your diamond?
7
   *
  ***
 *****
*******
 *****
  ***
   *
```



```
How many carats is your diamond?
9
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```