University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

AUTUMN TRIMESTER EXAMINATIONS
ACADEMIC YEAR 2020–2021

COMP 10280

Programming I (Conversion)

Professor Simon Thompson

Professor Chris Bleakley

Professor John Dunnion*

Time allowed: 90 minutes.

Instructions for Candidates

Answer both questions.

Students should write programs in Python 3.
Pseudocode for algorithms should be included with programs.

Students should upload their code to Brightspace every 15 minutes.

1. A palindrome is a string that reads the same backwards as it does forwards. For the purposes of this exercise, letters, numeric digits (0–9) and spaces are considered, but all other characters are ignored. Also, the case (uppercase/lowercase) of letters *is* to be considered. Thus "abba", "A-9*A" and "A2␣3b4b3␣2A" are palindromes according to this definition, but "Abba", "abc98cba" and "Madam,␣I'm␣Adam" are not.

   Write a program in Python 3 that prompts the user for a series of strings and checks whether each one is a palindrome according to the above definition. The program should continue until an empty string is entered. Your program should include a *non-recursive* function that checks whether a given string is a palindrome.

   Sample output from this program is as follows:

```
Enter a string (empty string to exit):  abba
abba is a palindrome.
Enter a string (empty string to exit):  A-9*A
A-9*A is a palindrome.
Enter a string (empty string to exit):  A2 3b4b3 2A
A2 3b4b3 2A is a palindrome.
Enter a string (empty string to exit):  Abba
Abba is not a palindrome.
Enter a string (empty string to exit):  abc98cba
abc98cba is not a palindrome.
Enter a string (empty string to exit):  Madam, I'm Adam
Madam, I'm Adam is not a palindrome.
Enter a string (empty string to exit):
Finished!
```

   [30 marks]

2. Write a program in Python 3 that prompts the user for a series of strings and checks whether each string or a prefix of the string (to a minimum length) is to be found in a file containing lines of text.

   The program should prompt the user for a file through which it should search. The program should then prompt the user for a string for which to search. If the string is found, the lines containing the string should be printed out and a message indicating that the string has been found should also be printed out. If the string is not found, the program should continue searching for prefixes of the string until either a prefix is found or the minimum prefix length (which is five) is reached. For example, if the user searches for the string "computability", the program will first search for "computability". If this is not found, the program will succesively search for "computabilit", "computabili", "computabil", "computabi", "computab", "computa", "comput" and "compu" until it finds a match. If no match is found, the program prints out a message to that effect.

   The program should continue until an empty string is entered.

   Notes:

   - Case (uppercase/lowercase) should be considered when searching for a string.
   - The minimum prefix length should be set to five, ie when the program is successively searching for prefixes of a string after an unsuccessful search, it should only search for a prefix the length of which is at least five.
   - When searching for prefixes of a given string, the program should stop searching once a prefix is found.
   - The program should always search for a given string, even if that string is shorter than the minimum length.
   - If the user does not specify a filename, the default filename is "lines.txt".
   - The program should check for the existence of the search file in the current working directory/folder.
   - A sample `lines.txt` file is available for download on the Brightspace site (thanks to Wikipedia and Cisco for the content!).

Sample output from this program is as follows (some debugging output is provided for extra information):

```
Enter a file to operate on (default is lines.txt): mylines.txt

Enter a string to search for (empty string to exit): programming
A computer is a machine that can be instructed to carry out sequences of arithmetic
or logical operations automatically via computer programming.
The purpose of programming is to find a sequence of instructions that will automate
the performance of a task (which can be as complex as an operating system) on a
computer, often for solving a given problem.
The process of programming thus often requires expertise in several different
subjects, including knowledge of the application domain, specialized algorithms,
and formal logic.
The search string programming or a prefix of it was found.

Enter a string (empty string to exit): programmable
Didn't find programmable. Now searching for prefixes...
Searching for: programmabl   length of s: 11   minlen:  5
Searching for: programmab   length of s: 10   minlen:  5
Jumpstart your journey into network programmability with this expert-led video
course.
The search string programmable or a prefix of it was found.

Enter a string (empty string to exit): cookiemonster
Didn't find cookiemonster. Now searching for prefixes...
Searching for: cookiemonste   length of s: 12   minlen: 5
Searching for: cookiemonst   length of s: 11   minlen: 5
Searching for: cookiemons   length of s: 10   minlen: 5
Searching for: cookiemon   length of s: 9   minlen: 5
Searching for: cookiemo   length of s: 8   minlen: 5
Searching for: cookiem   length of s: 7   minlen: 5
Searching for: cookie   length of s: 6   minlen: 5
Searching for: cooki   length of s: 5   minlen: 5
Neither the search string cookiemonster nor any of its prefixes of it was found.

Enter a string (empty string to exit):
Finished!
```

[70 marks]