

COP-4338 Systems Programming

Programming Assignment 1

Knight Foundation School of Computing and Information Sciences

In this assignment, you are asked to implement a program that recognizes interesting patterns for any given string. In an infinite loop, the program receives a single line from standard input (you may use `fgets(line, 1000, stdin)` to receive such line where `line` is a char array of length 1000). The program must be interested in finding one or more of the following special patterns in the input string:

- I. Singleton:** A singleton string is made of only one letter. Examples: `mmmmm`, `qqqqqqq`, `rr`, `s`, `yyy`
- II. Arithmetic:** A string made of subsequent alphabetical letters that appear in the alphabetical order. Examples: `bcdef`, `pqrstuvw`, `jk`, `y`
- III. Reverse Arithmetic:** A string made of subsequent alphabetical letters that appear in the reverse alphabetical order. Example: `fedcb`, `xwvutsrq`, `kj`, `y`
- IV. Balanced Tripartite:** A string made of three identical parts. Example: `busbusbus`, `laptoplaptoplaptop`, `zzz`
- V. Balanced Bipartite:** A string made of two identical halves. Examples: `ticktick`, `hophop`, `tantan`, `nocknock`, `nn`
- VI. Palindrome:** A palindrome reads the same backward as forward. Examples: `abcba`, `bob`, `g`

Please note that the above list is ranked in the decreasing order of their rarity. You program prints out the list of *all* patterns (that can be found for each input string) followed by a new line. Then, it asks for another line from input stream in a repetitive fashion. If no pattern is found for an input string, the program prints a new line only.

For example, consider the following string received from the input stream:

`bbbbbb\n`

This string is a singleton, balanced tripartite, balanced bipartite, and a palindrome. Therefore, the program must print the following:

`singleton\nbalanced tripartite\nbalanced bipartite\npalindrome\n\n`

Submissions

You need to submit a *.zip* file conmpressing the following files:

- the C source file(s) related to the assignment (*.c* files)
- header files (*.h* files)
- the makefile **Please use “`pattern_finder`” to name the executable of your program in the makefile.** Not including the makefile, or submitting a wrong one will reduce your score by 20%.
- a snapshot of your device running gdb on the program. (*.png* or *.jpg* files only)