

Assignment: Write a program in python that determines if a string contains a balanced set of brackets. Brackets consist of the pairs (), [], and { }. A string is a sequence of characters containing no white space. White space is a sequence of one or more blank characters, new line characters, or tab characters. A string is balanced if an opening bracket, (, [, or { is matched by the corresponding closing bracket,),], or }). Brackets are matched in a last-in-first-out order. If an opening curly brace, {, appeared in the string then the next bracket in the string must be a closing curly brace }. Any sequence of characters that are not brackets can appear between the opening and closing brackets.

Program Files: Project 3 consists of files **p03.py** and **Stack03.py**. Project file names are exactly as given. Failure to employ the foregoing names will result in a score of **zero (0)** for this project

Project files must be stored in the **root directory of your student account**. Failure to store project files in the root directory of your student account will result in a score of **zero (0)** for this project.

File	Description
p03.py	File p03.py contains functions that process command line arguments and distinguish strings having balanced brackets.

Stack03.py	File Stack03.py contains the implementation class Stack . Project 3 can be invoked with zero, one, or two program parameters. The first program parameter is the input file name. The second parameter is the output file name. Sample command lines together with corresponding actions by program p03 are shown below. Boldfaced type indicates data entered at the keyboard by the user.
-------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

\$ python3 p03.py

Enter the input file name: **i03.dat**

Enter the output file name: **o03.dat**

\$ python3 p03.py i03.dat

Enter the output file name: **o03.dat**

\$ python3 p03.py i03.dat o03.dat

Input File: File **i03.dat** in the class directory **~tt/cs4023/** contains a list of representative identifiers. Refer to Figure 1. Input file format.

Output File: Program **p03** produces file **o03.dat**. File **o03.dat** shown in Figure 2 is the output produced by program **p03** given the input file shown in Figure 1.

Figure 1.
Input file format

(({}{{(){}}}))
(({}{{(){}}}))
(o(t{b}m[s{t(f)s{t}d}f]s)p)us
(l(l{e}e[a{m(a)h{t}f]j]k)s)x

Figure 2.

Output file format

(({}{{(){}}})) is balanced.
(({}{{(){}}})) is not balanced.
(o(t{b}m[s{t(f)s{t}d}f]s)p)us is balanced.
(l(l{e}e[a{m(a)h{t}f]j]k)s)x is not balanced.