APL Assignment
Code Editor: Visual Studio Code
Language: Python 3
CECS 524 Unit 1 Assignment

Write, in the programming language of your choice, an interpreter for the language Brainfu*k. This is a Turing complete language. It is very simple, on the order of a late first semester of programming

Run it with this input below - what is the output?

++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>>.<-.<.
+++.------.--------.>>+.>++..

Source Code

```
# Brainfu*k Interpreter in Python CSULB APL Assignment CECS 524 Unit 1

'''
Author: Ishank Sharma
Date: 9 Sep 2024
References:
- https://en.wikipedia.org/wiki/Brainfuck - for overall understanding
- https://jwodder.freeshell.org/brainf.html - for wrap around knowledge
- https://minond.xyz/brainfuck/ - for visualising how loops work
'''

def interpret(code):
    memory_tape = [0] * 1000  # tape of size 1000
    ptr_tape = 0  # pointer on the tape
    i = 0  # another pointer to read the string
    bracket_mapper = {}  # Bracket map for jumps
    input_buffer = []  # Buffer to handle multi-character input
    # Bracket map to handle matching '[' and ']', using map for O(1) fast
retrieval instead of nested loop
    stack = []
    for index, j in enumerate(code):
        if j == '[':
            stack.append(index)
        elif j == ']':
            start = stack.pop()
            bracket_mapper[start] = index
            bracket_mapper[index] = start

    while i < len(code):
        if code[i] == '>':
```

```
            ptr_tape = (ptr_tape + 1) % 1000  # Prevents going out of
bounds
        elif code[i] == '<':
            ptr_tape = (ptr_tape - 1) % 1000
        elif code[i] == '+':
            memory_tape[ptr_tape] = (memory_tape[ptr_tape] + 1) % 256  #
Ensures memory value is between 0-255 for wrap arounds
        elif code[i] == '-':
            memory_tape[ptr_tape] = (memory_tape[ptr_tape] - 1) % 256
        elif code[i] == '.':
            print(chr(memory_tape[ptr_tape]), end='')
        elif code[i] == ',':
            memory_tape[ptr_tape] = ord(input()[0])
        elif code[i] == '[':
            # move to end of matching closed bracket if pointer is
pointing to zero bytes
            if memory_tape[ptr_tape] == 0:
                i = bracket_mapper[i]
        elif code[i] == ']':
            # move to starting of matching open bracket if pointer is
pointing to non zero bytes
            if memory_tape[ptr_tape] != 0:
                i = bracket_mapper[i]
        i += 1

if __name__ == "__main__":
    code = input("Welcome to the BrainFu*k Interpreter\nPlease Enter the
Brainfu*k code: ")
    # Sample Test

#++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>
>.<-.<.+++.------.--------.>>+.>++.
    print("Interpreter Output:")
    interpret(code)
```

Steps to run:

- Install python3

- **Paste the code and save as** `program.py`
- **Run the program using** `python3 program.py`

Input:

```
++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>>
.<-.<.+++.------.--------.>>+.>++.
```

Output: Hello World!

Snapshot of the output, refer figure below.