“””Простой интерпретатор элементарного текста браинфака”””

"""

Easy translator of BRAINFUCK

Hello World!

"""

brainfuck\_string = '+++++++++++++++++++++++++++++++++++++++++++++\

+++++++++++++++++++++++++++.+++++++++++++++++\

++++++++++++.+++++++..+++.-------------------\

---------------------------------------------\

---------------.+++++++++++++++++++++++++++++\

++++++++++++++++++++++++++.++++++++++++++++++\

++++++.+++.------.--------.------------------\

---------------------------------------------\

----.-----------------------.'

my\_num = 0

my\_list = []

my\_str = ''

for symbol\_in\_str in brainfuck\_string:

if symbol\_in\_str == '+':

my\_num += 1

elif symbol\_in\_str == '-':

my\_num -= 1

elif symbol\_in\_str == '.':

my\_str += chr(my\_num)

# print(chr(my\_num))

print(my\_str)

###########################################################################################################################################################################################################################

"""Более сложный интерпретатор"""

"""

Translator of BRAINFUCK

Hello World!

14.09.2021

"""

# version 1

brainf\_string = '++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++\

.>+.+++++++..+++.>++.<<+++++++++++++++.>.+++.\

------.--------.>+.>.'

'''

brainf\_string = '++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>>.<-.<.+++.------.--------.>>+.>++.'

'''

def do\_loop( fu\_brainf\_string, fu\_symbol\_list, fu\_index\_in\_brainf\_string, fu\_inddex\_in\_symbol\_list ):

left\_square\_bracket\_index = fu\_index\_in\_brainf\_string

loop\_symbol\_list\_index = fu\_inddex\_in\_symbol\_list

while fu\_symbol\_list[loop\_symbol\_list\_index]:

fu\_index\_in\_brainf\_string += 1

print("in fu 1")

if fu\_brainf\_string[fu\_index\_in\_brainf\_string] == "+":

fu\_symbol\_list[fu\_inddex\_in\_symbol\_list] += 1

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == "-":

fu\_symbol\_list[fu\_inddex\_in\_symbol\_list] -= 1

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == ">":

fu\_inddex\_in\_symbol\_list += 1

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == "<":

fu\_inddex\_in\_symbol\_list -= 1

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == ".":

print(fu\_symbol\_list[fu\_inddex\_in\_symbol\_list])

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == "[":

print("in fu-2")

fu\_brainf\_string, fu\_symbol\_list, fu\_index\_in\_brainf\_string, fu\_inddex\_in\_symbol\_list = do\_loop( fu\_brainf\_string, fu\_symbol\_list, fu\_index\_in\_brainf\_string, fu\_inddex\_in\_symbol\_list )

elif fu\_brainf\_string[fu\_index\_in\_brainf\_string] == "]": # возвращает в начало ветвления, то есть сразу в ячейку после знак "["

right\_square\_bracket\_index = fu\_index\_in\_brainf\_string

fu\_index\_in\_brainf\_string = left\_square\_bracket\_index

# fu\_index\_in\_brainf\_string = left\_square\_bracket\_index + 1

# fu\_index\_in\_brainf\_string += 1

return fu\_brainf\_string, fu\_symbol\_list, fu\_index\_in\_brainf\_string, fu\_inddex\_in\_symbol\_list

my\_num = 0

my\_list = []

my\_str = ''

symbol\_list = [0 for i in range(brainf\_string.count('>'))] # список члены которого будут ячейками с записанным в них значением выглядит так: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

index\_in\_brainf\_string = 0 # счет и обозначение на каком из элементов строки с символами мы находимся

inddex\_in\_symbol\_list = 0

print("symbol\_list at start:",symbol\_list)

print("len(brainf\_string):",len(brainf\_string))

print("brainf\_string.count('>')", brainf\_string.count('>'))

print("brainf\_string.count('<')", brainf\_string.count('<'))

#print("type(brainf\_string[17])", type(brainf\_string[17]))

while True:

if brainf\_string[index\_in\_brainf\_string] == "+":

symbol\_list[inddex\_in\_symbol\_list] += 1

elif brainf\_string[index\_in\_brainf\_string] == "-":

symbol\_list[inddex\_in\_symbol\_list] -= 1

elif brainf\_string[index\_in\_brainf\_string] == ">":

inddex\_in\_symbol\_list += 1

elif brainf\_string[index\_in\_brainf\_string] == "<":

inddex\_in\_symbol\_list -= 1

elif brainf\_string[index\_in\_brainf\_string] == ".":

my\_str += chr(symbol\_list[inddex\_in\_symbol\_list])

print(chr(symbol\_list[inddex\_in\_symbol\_list]))

elif brainf\_string[index\_in\_brainf\_string] == "[":

brainf\_string, symbol\_list, index\_in\_brainf\_string, inddex\_in\_symbol\_list = do\_loop( brainf\_string, symbol\_list, index\_in\_brainf\_string, inddex\_in\_symbol\_list )

index\_in\_brainf\_string += 1 # после всех сравнений пройти к следующему значению в списке "brainf\_string"

if index\_in\_brainf\_string == len(brainf\_string): # выйти из ветвления если дошли до конца списка "brainf\_string"

break

print("The strin is: ", my\_str)

print("symbol\_list at end:",symbol\_list)