A Nightmare On Math Street

Enum

```
┌──(root㉿kali)-[~]
└─# nc 165.22.115.189 31444

#####################################################################
#                                                                   #
# I told you not to fall asleep!                                    #
#                                                                   #
# A 500 question quiz is coming up.                                 #
#                                                                   #
# Be careful; Dream math works a little differently:                #
# Addition and multiplication have the REVERSE order of operation.  #
#                                                                   #
# And remember, if you fail in your sleep, you fail in real life... #
#                                                                   #
#####################################################################


[001]: 51 + 87 * 59 * 54 + 43 = ?
```

> Addition and multiplication have the REVERSE order of operation.

Ok, looks like I will have to write an algorithm to adjust the operators

I gathered samples again and again to make the logic clear

Example:

```
78 + 33 + 91 + 54 * 95 * 73
```

will be ->

```
(78 + 33 + 91 + 54) * 95 * 73
```

Script

My Script:

```
from pwn import *
import re


def adjust_operators(question: str) -> str:
    """Adjust the question to to make Addition and multiplication have the REVERSE order of operation"""
    question_adjusted = []
    during_plus_op = False
    for unit in question.split():
        if unit == "*":
            if during_plus_op:
                question_adjusted[-1] = f"{question_adjusted[-1]}) *"
                during_plus_op = False
            else:
                question_adjusted.append(unit)
            continue

        if unit == "+":
            # Check if next unit contains "("
```

```
            if question[question.index(unit)+1].startswith("("):
                question_adjusted.append(unit)
                continue
            if during_plus_op:
                question_adjusted[-1] = f"{question_adjusted[-1]} +"
            else:
                question_adjusted[-1] = f"({question_adjusted[-1]} +"
                during_plus_op = True
            continue

        question_adjusted.append(unit)

    if during_plus_op:
        question_adjusted[-1] += ")"

    question_adjusted_str = " ".join(question_adjusted)
    print(f"{question_adjusted_str=}")
    return question_adjusted_str

def solve_quesetion():
    """Solve the question and send the answer"""
    question_str = conn.recvuntil("?", timeout=1).decode()
    print(f"{question_str=}")
    question = re.findall(r"\[\d+\]:\s+(.*?)\s+\=", question_str)[0]
    print(f"{question=}")
    question_adjusted = adjust_operators(question)
    answer = str(eval(question_adjusted))
    print(f"{answer=}")

    conn.sendline(answer)
    print("\n---\n")
    if "[500]" in question_str:
        print(conn.recvline_contains("HTB", timeout=1).decode())
```

```
        exit(0)


global conn
conn = remote('165.227.237.190', 31344)

while 1:
    try:
        solve_quesetion()
    except EOFError:
        print("EOFError...")
        break
```

Better solution by my friend:

```
import socket
import re

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("IP",PORT))

def checkFor(toEval,regex):
    while re.findall(regex,toEval):
        found=re.findall(regex,toEval)[0]
        toEval=toEval.replace(found,str(eval(found)),1)
    return toEval

def evaluate(content):
    while len(re.findall(r"[0-9]{1,99}",content))>1:
        if(re.findall(r"\([0-9*+ ]*\)",content)):
            nextContent=re.findall(r"\([0-9*+ ]*\)",content)[0]
```

```
            content=content.replace(nextContent,evaluate(nextContent[1:-1]))
        else:
            while len(re.findall(r"[0-9]{1,99}",content))>1:
                content=checkFor(content,r"[0-9]{1,99} \+ [0-9]{1,99}")
                content=checkFor(content,r"[0-9]{1,99} \* [0-9]{1,99}")
    return content

for nb in range(500):
    print(f"{nb+1}/500")
    reply = str(s.recv(4096))
    toEval=re.findall(r"]: (.+)? =",reply)[0]
    toEval=evaluate(toEval)
    result=eval(toEval)
    s.sendall(bytes(f"{result}\n","utf-8"))

flag = str(s.recv(4096))
print(flag)
```

Flag

```
---

question_str='\n> [500]: (99 * (28 * 74) + 29) = ?'
question='(99 * (28 * 74) + 29)'
question_adjusted_str='(99 * (28 * (74) + 29))'
answer='207999'

---

> Well done! Here's the flag: HTB{tH0s3***********_5k1llz}
[*] Closed connection to 165.227.237.190 port 31344
```

This is the first time I got <100 RANK on challenges/machines ^^