Samet Cagri Aktepe – 12.06.2022

https://www.spoj.com/problems/ONP/

Transform the algebraic expression with brackets into RPN form (Reverse Polish Notation). Two-argument operators: +, -, *, /, ^ (priority from the lowest to the highest), brackets ( ). Operands: only letters: a,b,...,z. Assume that there is only one RPN form (no expressions like a*b*c).

Input

```
t [the number of expressions <= 100]

expression [length <= 400]

[other expressions]
```

Text grouped in [ ] does not appear in the input file.

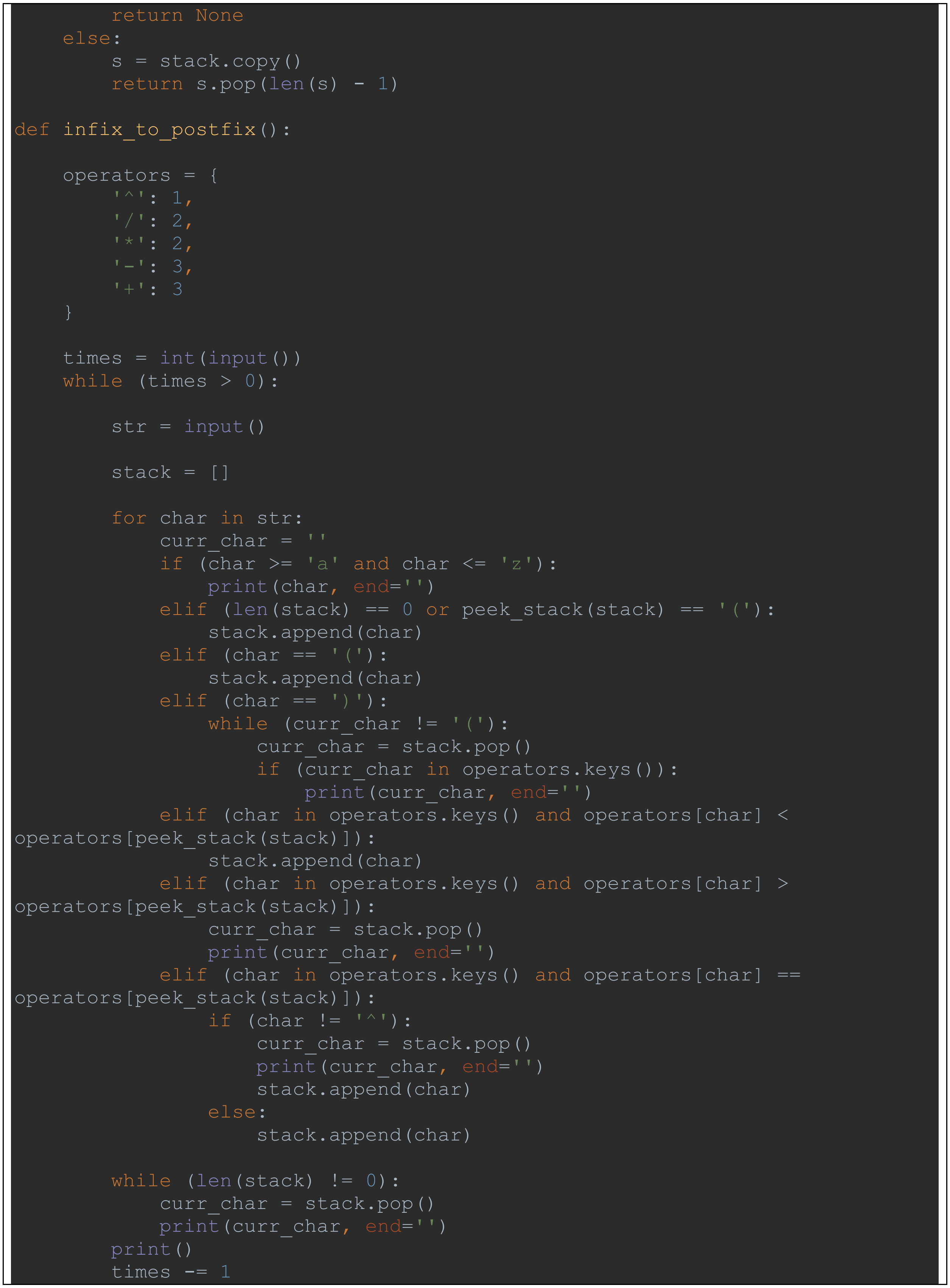
Output

```
The expressions in RPN form, one per line.
```

Example

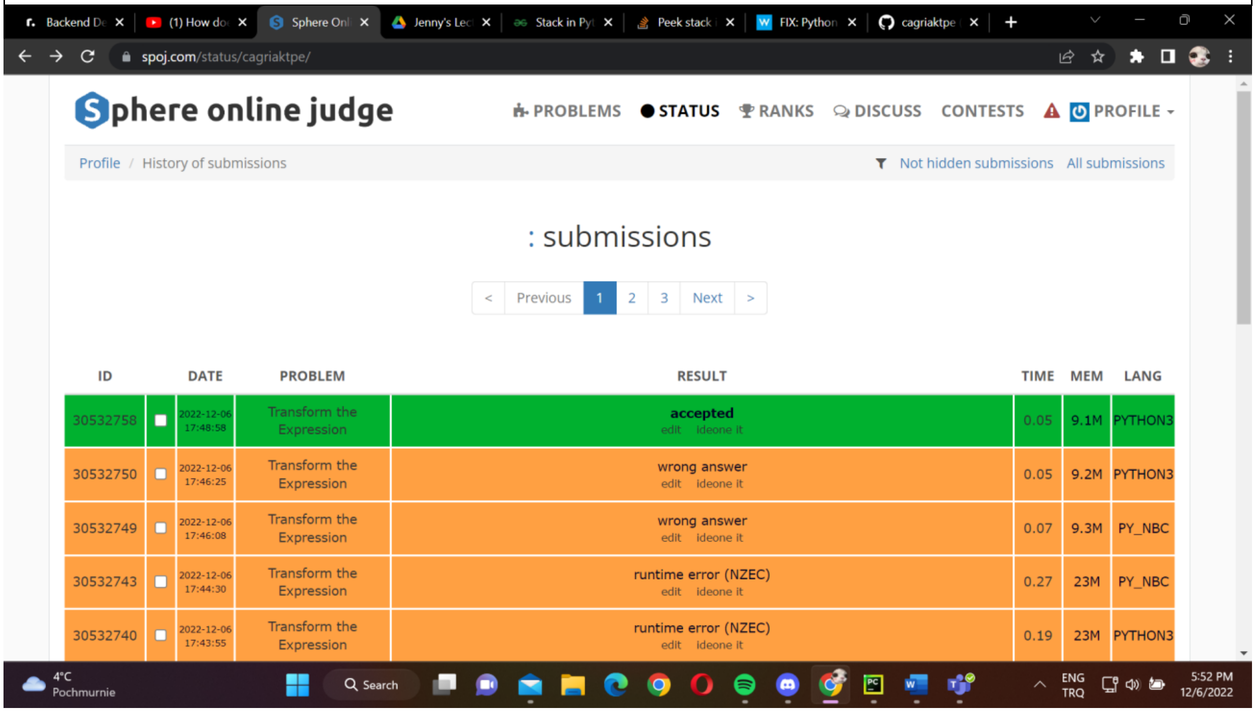
```
Input:

3

(a+(b*c))

((a+b)*(z+x))

((a+t)*((b+(a+c))^(c+d)))



Output:

abc*+

ab+zx+*

at+bac++cd+^*
```

```
def peek_stack(stack):
    if stack == []:
```



```
        return None
    else:
        s = stack.copy()
        return s.pop(len(s) - 1)

def infix_to_postfix():

    operators = {
        '^': 1,
        '/': 2,
        '*': 2,
        '-': 3,
        '+': 3
    }

    times = int(input())
    while (times > 0):

        str = input()

        stack = []

        for char in str:
            curr_char = ''
            if (char >= 'a' and char <= 'z'):
                print(char, end='')
            elif (len(stack) == 0 or peek_stack(stack) == '('):
                stack.append(char)
            elif (char == '('):
                stack.append(char)
            elif (char == ')'):
                while (curr_char != '('):
                    curr_char = stack.pop()
                    if (curr_char in operators.keys()):
                        print(curr_char, end='')
            elif (char in operators.keys() and operators[char] <
operators[peek_stack(stack)]):
                stack.append(char)
            elif (char in operators.keys() and operators[char] >
operators[peek_stack(stack)]):
                curr_char = stack.pop()
                print(curr_char, end='')
            elif (char in operators.keys() and operators[char] ==
operators[peek_stack(stack)]):
                if (char != '^'):
                    curr_char = stack.pop()
                    print(curr_char, end='')
                    stack.append(char)
                else:
                    stack.append(char)

        while (len(stack) != 0):
            curr_char = stack.pop()
            print(curr_char, end='')
        print()
        times -= 1
```

```
infix_to_postfix()
```



Profile / History of submissions

Not hidden submissions All submissions

: submissions

< Previous 1 2 3 Next >

ID		DATE	PROBLEM	RESULT	TIME	MEM	LANG
30532758	□	2022-12-06 17:48:58	Transform the Expression	**accepted** edit ideone it	0.05	9.1M	PYTHON3
30532750	□	2022-12-06 17:46:25	Transform the Expression	wrong answer edit ideone it	0.05	9.2M	PYTHON3
30532749	□	2022-12-06 17:46:08	Transform the Expression	wrong answer edit ideone it	0.07	9.3M	PY_NBC
30532743	□	2022-12-06 17:44:30	Transform the Expression	runtime error (NZEC) edit ideone it	0.27	23M	PY_NBC
30532740	□	2022-12-06 17:43:55	Transform the Expression	runtime error (NZEC) edit ideone it	0.19	23M	PYTHON3