Prefix Evaluation And Conversions

1. You are given a prefix expression.
2. You are required to evaluate it and print it's value.
3. You are required to convert it to infix and print it.
4. You are required to convert it to postfix and print it.
Note -> Use brackets in infix expression for indicating precedence. Check sample input output for more details.
Input Format

Input is managed for you

Output Format

value, a number
infix
prefix
Constraints

1. Expression is a valid prefix expression
*2. The only operators used are +, -, , /
3. All operands are single digit numbers.

Sample Input

*-+2/6483

Sample Output

2
((2+((6*4)/8))-3)
264*8/+3-
Bas isme logic yeh hai ki hume back se traverse karna hota hai hai and first pop is first element itself unlike in last 2 notations where first pop was
second element

```
def preToInPost(string):
    post = []
    infix = []

    for i in range(len(string)-1,-1,-1):
        ch = string[i]
        if ch in {'+','-','*','/'}:
            inv1 = infix.pop()
            inv2 = infix.pop()
            temp1 = '('+inv1+ch+inv2+')'
            infix.append(temp1)

            pos1 = post.pop()
            pos2 = post.pop()
            temp2 = pos1+pos2+ch
            post.append(temp2)
        else:
            post.append(ch)
            infix.append(ch)
    return post[-1],infix[-1]


string = '-+2/*6483'

print(preToInPost(string))
```