Prefix to Postfix

Prefix exp ⇒ +-*AB+CDE → input

infix exp → A*B-(C+D)+E

→ A*B − CD+ +E
→ AB* − CD+ +E
→ AB*CD+− +E
→ AB*CD+−E+ → postfix → output

op2+op1+ch

op1 = C
op2 = D
op1+op2+ch E

op1 B
op2 A B
2+1 A

Algorithm for Prefix to postfix ⇒

① Read the prefix expression from right to left.

```
Stack<String> stack = new Stack();
for(int i= prefix.length()-1; i>=0; i--){
    char ch = prefix.charAt(i);
}
```

② If incoming symbol is operand then push into the stack.

```
Stack<String> stack = new Stack<>();
for(int i= prefix.length-1; i>=0; i--){
    char ch= prefix.charAt(i);
    if( isOperand(ch)){
        stack.push(ch+"");
    }
}
```

```
public static boolean isOperand(char ch){
    return (ch>='a' && ch<='z') ||
           (ch>='A' && ch<='Z');
}
```

③ in incoming symbol is an operator then pop two element from the stack, create a string by concatenating the two operand and operator after them. and resultant string push into the stack.

```
String temp = operand1 + operand2 + ch(operator)
stack.push(temp);
```

④ Repeat the above steps until end of the prefix exp.

⑤ then finally return the pop element from the stack.

program for Prefix to postfix ⇒

prefix ⇒ +-*AB+CDE

9-1=i

i = 8 7 6 5 4 3 2 1 0

char ch = prefix.charAt(i);

E D C + B A * − + → AB*CD+−

String op1 = stack.pop(); // A AB*

String op2 = stack.pop(); // B CD+ E

string temp = op1 + op2 + ch(operator); → CD+

stack.push(temp) →

AB*

AB*CD+−

AB*CD+−E+

Stack: A, C, D, E

B* AB*

CD+ − AB*CD+

− AB*CD+−E+

top = -1 0 1 2 1 0 1 2 3 2 1 2 1 0 1 0 −1 0

finally stack have only one element which is resultant string.

then return the popped element from the stack.

```
return stack.pop();
```

```
public class prefixToPostfix{
    public static boolean isOperand(char ch){
        return (ch>='a' && ch<='z')||(ch=>'A' && ch<='Z'));
    }
    public static String prefixToPostfix(String prefix){
        Stack<String> stack = new Stack<>();
        for(int i= prefix.length()-1; i>=0; i--){
            char ch= prefix.charAt(i);
            if( isOperand(ch)){
                stack.push(ch));
            }else{
                String operand1 = stack.pop();
                String operand2 = stack.pop();
                String temp = operand1 + operand2 + ch;
                stack.push(temp);
            }
        }
        return stack.pop();
    }
    public static void main(String arg[]){
        String prefix = +-*AB+CDE;
        sop("Prefix :"+ prefix);
        String postfix = prefixtopostfix(infix);
        sop("postfix :"+ postfix);
    }
}
```

25-08-2023 covered topic

① prefix to postfix conversion.

26-08-2023 topic

① Postfix to prefix expression

② postfix to infix expression