Обратная польская нотация (ОПН) — форма записи математических и логических

выражений, в которой операнды расположены перед знаками операций.

Алгоритм Дейкстры – перевод числа из инфиксной нотации в ОПН.

Особенности обратной польской записи следующие:

• Порядок выполнения операций однозначно задаётся порядком следования знаков

операций в выражении, поэтому отпадает необходимость использования скобок и

введения приоритетов и ассоциативности операций.

• В отличие от инфиксной записи, невозможно использовать одни и те же знаки для

записи унарных и бинарных операций. Так, в инфиксной записи выражение 5 \* (-

3 + 8) использует знак «минус» как символ унарной операции (изменение знака

числа), а выражение (10 - 15) \* 3 применяет этот же знак для обозначения

бинарной операции (вычитание). Конкретная операция определяется тем, в какой

позиции находится знак. Обратная польская запись не позволяет этого: запись 5 3

- 8 + \* (условный аналог первого выражения) будет интерпретирована как

ошибочная, поскольку невозможно определить, что «минус» после 5 и 3

обозначает не вычитание; в результате будет сделана попытка вычислить сначала 5

- 3, затем 2 + 8, после чего выяснится, что для операции умножения не хватает

операндов. Чтобы всё же записать это выражение, придётся либо

переформулировать его, либо ввести для операции изменения знака отдельное

обозначение, например, «±»: 5 3 ± 8 + \*.

Так же, как и в инфиксной нотации, в ОПН одно и то же вычисление может быть

записано в нескольких разных вариантах. Например, выражение (10 - 15) \* 3 в

ОПН можно записать как 10 15 - 3 \*, а можно — как 3 10 15 - \*

Из-за отсутствия скобок обратная польская запись короче инфиксной. За этот счёт

при вычислениях на калькуляторах повышается скорость работы оператора

(уменьшается количество нажимаемых клавиш), а в программируемых устройствах

сокращается объём тех частей программы, которые описывают вычисления.

Последнее может быть немаловажно для портативных и встроенных

вычислительных устройств, имеющих жёсткие ограничения на объём памяти.

Пример:

**Вход: 3 + 4 \* 2 / (1 - 5)^2**

Читаем «3»

Добавим «3» к выходной строке

Выход: 3

Читаем «+»

Кладём «+» в стек

Выход: 3

Стек: +

Читаем «4»

Добавим «4» к выходной строке

Выход: 3 4

Стек: +

Читаем «\*»

Кладём «\*» в стек

Выход: 3 4

Стек: + \*

Читаем «2»

Добавим «2» к выходной строке

Выход: 3 4 2

Стек: + \*

Читаем «/»

Выталкиваем «\*» из стека в выходную строку, кладём «/» в стек

Выход: 3 4 2 \*

Стек: + /

Читаем «(»

Кладём «(» в стек

Выход: 3 4 2 \*

Стек: + / (

Читаем «1»

Добавим «1» к выходной строке

Выход: 3 4 2 \* 1

Стек: + / (

Читаем «-»

Кладём «-» в стек

Выход: 3 4 2 \* 1

Стек: + / ( -

Читаем «5»

Добавим «5» к выходной строке

Выход: 3 4 2 \* 1 5

Стек: + / ( -

Читаем «)»

Выталкиваем «-» из стека в выходную строку, выталкиваем «(»

Выход: 3 4 2 \* 1 5 -

Стек: + /

Читаем «^»

Кладём «^» в стек

Выход: 3 4 2 \* 1 5 -

Стек: + / ^

Читаем «2»

Добавим «2» к выходной строке

Выход: 3 4 2 \* 1 5 - 2

Стек: + / ^

Конец выражения

Выталкиваем все элементы из стека в строку

**Выход: 3 4 2 \* 1 5 - 2 ^ / +**