

Odwrotna Notacja Polska (ONP)

Odwrotna Notacja Polska (ONP) to system notacji beznawiasowej umożliwiający zapis wyrażeń arytmetycznych w ten sposób, że najpierw występują operandy (argumenty) a następnie operator (działanie).

Np.

2 3 + (ONP)	odpowiada wyrażeniu	2+3
x sin (ONP)	odpowiada wyrażeniu	sin x
x y + sin (ONP)	odpowiada wyrażeniu	sin(x+y)
a~b+c/de-*	odpowiada wyrażeniu	(-a+b)/c*(d-e)

Dla wyrażenia zapisanego w ONP obliczenia są wykonywane od lewej w kolejności występowania operatorów.

ONP jest jedną z najczęściej używanych form pośrednich wykorzystywanych przy konstrukcji kompilatorów.

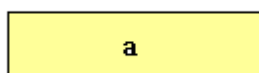
Obliczanie wartości wyrażenia w ONP –algorytm:

```
while (nie_koniec_danych)
{
    pobierz_kolejny_element;
    if (element_jest_operandem)
        wstaw_wartość_operanda_na_stos; //operacja PUSH
    Else //element jest operatorem
    {
        pobierz_ze_stosu_potrzebne_argumenty; //operacje TOP i POP
        wykonaj_działanie;
        wstaw_wynik_na_stos; //operacja PUSH
    }
} //wartość wyrażenia jest na szczycie stosu;
```

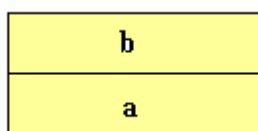
Przykład:

Wyrażenie w ONP : $a\ b\ +\ c\ d\ +\ *$

Wczytujemy **a** (operand) i wstawiamy na stos:

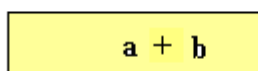


Wczytujemy **b** (operand) i wstawiamy na stos:



Wczytujemy **+** (operator) i :

- pobieramy argumenty {a i b} {stos staje się pusty}
- wykonujemy działanie {a+b}
- wynik wstawiamy na stos:



Wczytujemy **c** (operand) i wstawiamy na stos:

c
a + b

Wczytujemy **d** (operand) i wstawiamy na stos:

d
c
a + b

Wczytujemy **+** (operator) i :

- pobieramy argumenty {c i d} {na stosie zostaje a+b}
- wykonujemy działanie {c+d}
- wynik wstawiamy na stos:

c + d
a + b

Wczytujemy ***** (operator) i :

- pobieramy argumenty {a+b i c+d} {stos jest pusty}
- wykonujemy działanie {(a+b)*(c+d)}
- wynik wstawiamy na stos:

(a+b)*(c+d)

Zad. 1. Obliczyć wartości wyrażeń w ONP:

a) $a \ b \ c \ d \ * \ - \ / \ b \ 2 \ ^ \ +$

b) $a \ b \ c \ * \ - \ d \ b \ / \ 2 \ ^ \ +$

c) $a \ b \ c \ * \ - \ d \ b \ 2 \ / \ ^ \ +$

Konwersja wyrażenia z notacji tradycyjnej do ONP –algorytm

```
while (nie_koniec_danych)
{
    weź_kolejny_element;
    if (element_jest_operandem) dopisz_element_na_wyjście;
    Else //element jest operatorem_lub_nawiasem
    {
        if (element_jest_operatorem)
        while (priorytet_operatora_na_stosie >= priorytet_elementu)
            przepis_operator_ze_stosu_na_wyjście;
        wstaw_operator_na_stos; // operacja PUSH
    }
    Else
    If (element == "(, )" wstaw_element_na_stos;
    Else
    If (element == "),, )" //zdejmuj ze stosu aż do „,,”
    while (na_stosie_operator)
        przepis_operator_ze_stosu_na_wyjście; //op. POP

    zdejmij_ze_stosu_“(, //operacja POP
}
while (nie_pusty_stos)
    przepis_operator_ze_stosu_na_wyjście;
```

Przykład.2.
 $(a+b)*(c+d)$

Wejście	Stos	Wyjście
$(a+b)*(c+d)$	(
$a+b)*(c+d)$		a
$+b)*(c+d)$	(+	a
$b)*(c+d)$	(+	ab
$)*(c+d)$		ab+
$*(c+d)$	*	ab+
$(c+d)$	*(ab+
$c+d)$	*(ab+c
$+d)$	*(+	ab+c
$d)$	*(+	ab+cd
)		ab+cd+
		ab+cd+*

Zad.2.

- a)** $a/(b-c*d)+(b^2)$
- b)** $a-(b*c)+(d/b)^2$
- c)** $(a+(b-c*d))/(b^2)$
- d)** $(a-b*c)+d^{(b/2)}$