

13. Odwrotna notacja polska

1 Przekształcenie wyrażenia do ONP i obliczenie jego wartości

1.1 Odwrotna notacja polska

Odwrotna notacja polska (ONP, ang. reverse Polish notation, RPN) jest to sposób zapisu wyrażeń arytmetycznych, w którym znak wykonywanej operacji umieszczony jest po operandach (zapis postfiksowy), a nie pomiędzy nimi jak w konwencjonalnym zapisie algebraicznym (zapis infiksowy) lub przed operandami jak w zwykłej notacji polskiej (zapis prefiksowy). Zapis ten pozwala na całkowitą rezygnację z użycia nawiasów w wyrażeniach, jako że jednoznacznie określa kolejność wykonywanych działań.

ONP bardzo ułatwia wykonywanie na komputerze obliczeń z nawiasami i zachowaniem kolejności działań oraz ich priorytetów. Zarówno algorytm konwersji notacji konwencjonalnej (infiksowej) na odwrotną notację polską (postfiksową), jak i algorytm obliczania wartości wyrażenia danego w ONP są bardzo proste i wykorzystują stos.

Odwrotna notacja polska została opracowana przez australijskiego naukowca Charlesa Hamblina jako “odwrócenie” beznawiasowej notacji polskiej Jana Łukasiewicza na potrzeby zastosowań informatycznych.

1.2 Przekształcenie wyrażenia do ONP

Szablon programu należy uzupełnić o definicję funkcji `void to_onp(char* expression, char* onp)`. Łańcuch `expression` zawiera wyrażenie w postaci infiksowej. Po wykonaniu funkcji, łańcuch `onp` powinien zawierać to wyrażenie przekształcone do postaci postfiksowej (ONP).

Wyrażenie wejściowe jest poprawnym wyrażeniem arytmetycznym składającym się z liczb całkowitych, operatorów (+, -, *, /) i nawiasów okrągłych, wyznaczających kolejność wykonania operacji. Priorytet i łączność operatorów jest zdefiniowana jak w języku C.

1.3 Obliczenie wartości wyrażenia zapisanego w ONP

Szablon programu należy uzupełnić o definicję funkcji `int evaluate_onp(char* onp)`. Łańcuch `onp` zawiera wyrażenie w postaci ONP, otrzymane w wyniku wykonania poprzednio opisanej funkcji. Funkcja zwraca wartość podanego wyrażenia.

1.4 Specyfikacja

Program wczytuje linię tekstu, zawierającą wyrażenie w postaci infiksowej.

Każdy element wyrażenia (liczby, operatory, nawiasy) jest oddzielony przynajmniej jednym znakiem spacji lub tabulacji.

Na wyjściu program wypisuje postać postfiksową (ONP) wczytanego wyrażenia a w kolejnej linii jego wartość.

Podobnie jak w przypadku wyrażenia wejściowego, każdy element wyrażenia w ONP jest oddzielony znakiem spacji.

- **Wejście**
wyrażenie w postaci infiksowej
- **Wyjście**
wyrażenie w postaci postfiksowej (ONP)
wartość wyrażenia
- **Przykład**
Wejście:
 $12 * (8 - 4) / 3$
Wyjście:
 $12 8 4 - * 3 /$
16