

# Języki formalne i techniki translacji

## Laboratorium - lista nr 3

Termin oddania: ostatnie zajęcia przed 14 grudnia 2024

Zapoznaj się samodzielnie z dokumentacją i przykładami programu BISON (<http://www.gnu.org/software/bison/manual/>).

### Zadanie 1

Używając LEX-a i BISON-a zaimplementuj translator wyrażeń arytmetycznych w ciele  $GF(1234577)$  z postaci infiksowej do postaci postfiksowej (odwrotnej notacji polskiej), z korekcją postaci liczby (w  $Gp$  nie ma liczb ujemnych i większych lub równych  $p$ ), i podającej wynik obliczenia wyrażenia. Wyrażenia do policzenia umieszczone są w osobnych liniach. Program ma przetwarzać wszystkie linie wejścia, a linie zaczynające się od # traktować jak linie komentarza i omijać. W przypadku długich linii ma być możliwość ich podzielenie za pomocą znaku \ (tak jak w języku c).

Zadbaj o właściwe priorytety operatorów, właściwą łączność operatorów i odpowiednią obsługę błędów. Pamiętaj o unarnym operatorze - dla danych wejściowych (często dla wygody w  $GF(p)$  piszemy np.  $-1$  zamiast  $p - 1$ ). Potęgowanie powinno być dozwolone tylko jako operacja uogólniająca mnożenie - nie wolno składać potęg (można liczyć  $(a^b)^c$ , ale nie można  $a^{b^c}$ ).

Przykładowa sesja z napisanym programem może wyglądać następująco:

```
2+3*(4-5)
2 3 4 5 - * +
Wynik: 1234576
2^100
2 100 ^
Wynik: 295422
# ala ma kota
2-3-2
2 3 - 2 -
Wynik: 1234574
269164/123456
269164 123456 /
Wynik: 567890
-2--1
1234575 1234576 -
Wynik: 1234576
1/-580978
1 653599 /
Wynik: 123456
123456789
1233666
Wynik: 1233666
-1234567
10
Wynik: 10
2+3*(4-5
```

Błąd.

$2^{123}$

2 123 ^

Wynik: 594706

$2^{-2}$

2 1234574 ^

Wynik: 925933

## Zadanie 2

Napisz poprzednie zadanie w innym języku programowania, np. w Pythonie z pakietem `PLY` lub `SLY` albo Javie z biblioteką `ANTLR` (inne języki uzgodnij z prowadzącym laboratorium).