



Oppgave: grammatikk, parser (+ skanner og interpret) for språket:

$\text{exp} \rightarrow \text{exp op number} \mid \text{number}$

$\text{op} \rightarrow + \mid -$

1. Skriv om grammatikken til en ren BNF-grammatik uten venstre-rekursjon
2. Foreta evt. venstre-faktorisering
3. Finn First og Follow-mengdene
4. Lag $M[N,T]$ – tabellen
5. Lag i f.eks Java:
 1. En scanner for språket
 1. som kan lese språket token for token
bestemme type + evt. verdi : NUMBER, ADD eller SUBTR
(bruk f.eks pakken easyIO fra INF1000)
 2. En parser for språket
 1. bygger opp et syntakstre
 2. skriver ut treet på prefix form
 3. En interpret
 1. går gjennom treet og beregner verdien til uttrykket



To enkle eksempler på utskrift av kjøring:

1 - 2 og: 3 - 4 - 5

```
>java Tre 12.txt
```

```
Test av setning:1 - 2
```

```
(SUBTR:(NUMBER:1),(NUMBER:2))
```

```
Evaluering av venstre-tre av: 1 - 2 er: -1
```

```
> java Tre 345.txt
```

```
Test av setning: 3 - 4 - 5
```

```
(SUBTR:(SUBTR:(NUMBER:3),(NUMBER:4)),(NUMBER:5))
```

```
Evaluering av venstre-tre av: 3 - 4 - 5 er: -6
```


$$1 - 2 - 3 - 4 - 5 - 6 + 6 + 5 + 4 + 3 + 2$$

java Tre 1-6.txt

Test av setning: 1 - 2 - 3 - 4 - 5 - 6 + 6 + 5 + 4 + 3 + 2

(ADD:(ADD:(ADD:(ADD:(ADD:(SUBTR:(SUBTR:(SUBTR:(SUBTR:(SUBTR:(NUMBER:1),(NUMBER:2)),(NUMBER:3)),(NUMBER:4)),(NUMBER:5)),(NUMBER:6)),(NUMBER:6)),(NUMBER:5)),(NUMBER:4)),(NUMBER:3)),(NUMBER:2))

Evaluering av venstre-tre av: 1 - 2 - 3 - 4 - 5 - 6 + 6 + 5 + 4 + 3 + 2 er: 1