Exercise 1

Convert the following grammar to Chomsky normal form:

```
E -> E + E | E * E | ( E ) | id
```

Exercise 2

Convert the following grammar to Chomsky normal form:

```
S -> a S b | ε
```

Then, apply the CYK algorithm on the following inputs:

```
a a b b
a b b
```

Exercise 3

Convert the following grammar to Chomsky normal form:

```
S -> P ;
P -> I | I ; P
I -> if E then P R | print E
R -> else P | ε
W -> while E do P
E -> L | E or E | C
C -> C and E | E and C
L -> true | false
```

Then, apply the CYK algorithm on the following inputs:

```
if true print true ;
```

```
print true or false or true ;
```

```
if true then print true ; print false ;
```

In each case, how many different parse trees do you obtain ?

How many parse trees are there for the following inputs ?
You do not need to execute CYK to answer this question.

```
if true or false or true then print true or false or true ;
```

```
if true then if false then print true else print false ;
```

```
if true then print true ; print false ; print true ;
```