function CYK(words, grammar) returns best_parse

Create and clear \( p[num\_words, num\_words, num\_nonterminals] \)

# base case
for \( i = 1 \) to \( num\_words \)
    for \( A = 1 \) to \( num\_nonterminals \)
        if \( A \rightarrow w_i \) is in grammar then
            \( \pi[i, i, A] = P(A \rightarrow w_i) \)

# recursive case
for \( j = 2 \) to \( num\_words \)
    for \( i = 1 \) to \( num\_words - j + 1 \)
        for \( k = 1 \) to \( j - 1 \)
            for \( A = 1 \) to \( num\_nonterminals \)
                for \( B = 1 \) to \( num\_nonterminals \)
                    for \( C = 1 \) to \( num\_nonterminals \)
                        prob = \( \pi[i, k, B] \times p[i+k, j-k, C] \times P(A \rightarrow BC) \)
                        if \( prob > \pi[i, j, A] \) then
                            \( \pi[i, j, A] = prob \)
                            \( B[i, j, A] = \{k, A, B\} \)