Adam Cherek s464922

zadanie algorytmiczne nr 3:

algorytm będzie dla zadanego słowa P w postaci listy brał głowę listy, dodawał ją na początek listy rezultatu, a następnie wykonywał to samo dla ogona słowa P aż odwróci całe słowo

```
funkcja REV(słowo): zwraca odbicie zwierciadlane słowa
{
        list result := []                   % inicjujemy potrzebne listy %
        list P:= słowo
        if P = [] then                      % jeżeli słowo jest puste to zwracamy pustą listę %
                return result
        else                                % jeżeli nie jest puste to dokładamy kolejne głowy na
                                              początek listy aż ogon będzie pusty %
                result := MAKELIST(HEAD(P), result)
                while TAIL(P) != [] do
                        P := TAIL(P)
                        result := MAKELIST(HEAD(P), result)
                return result               % zwracamy odwrócone słowo po zakończeniu
                                              sprawdzania ogona %
}
```

zadanie algorytmiczne nr 4:

algorytm będzie dodawał za pomocą MAKELIST() kolejne głowy odbicia zwierciadlanego słowa P do słowa Q

```
funkcja CON(słowo, słowo2): zwraca słowo będące konkatenacją słów
{
        list P := REV(słowo)                    % inicjujemy potrzebne listy %
        list Q := słowo2
        if P = [] then              % sprawdzamy czy P nie jest puste %
                return Q
        else                                % jeżeli nie jest puste to dokładamy kolejne głowy na
                                              początek słowa Q aż ogon będzie pusty %
                Q := MAKELIST(HEAD(P), Q)
                while TAIL(P) != [] do
                        P := TAIL(P)
                        result := MAKELIST(HEAD(P), Q)
                return Q                    % zwracamy Q będące obecnie konkatenacją słów %

}
```