

zadanie algorytmiczne nr 7:

algorytm będzie dla danego słowa P w postaci listy oraz liczby i, przechodził po słowie dodając kolejne głowy listy do nowego słowa, pomijając i-tą głowę i zwracając nowe słowo niezawierające i-tego symbolu słowa P

funkcja REMOVESYMBOL(słowoP, liczbai): zwraca słowo będące P bez i-tego symbolu słowa P, wykorzystując funkcje LENGTH(P) oraz REV(P)

```
{
    int pos := 1                % inicjujemy potrzebne zmienne %
    list P:= słowoP
    if liczbai > LENGTH(słowoP) then          % sprawdzamy możliwe sprzeczności %
        return ERROR
    else
        % jeżeli nie ma sprzeczności to zaczynamy
        % przechodzić po głowach słowa P %
        list Q := []
        while pos < LENGTH(słowoP) do
            if pos != liczbai then
                MAKELIST(HEAD(P), Q)
                P:= TAIL(P)
                pos := pos + 1
        return REV(Q)          % zwracamy odwrotność Q aktualnie będące słowem P
                                % bez jego i-tego symbolu %
}
```

zadanie algorytmiczne nr 8:

wykorzystując zależność: jeżeli Q jest podciągiem P i P jest podciągiem Q to P=Q algorytm sprawdza przy pomocy SUBSTRING(Q,P) tę właśnie zależność i zwraca TAK lub NIE w zależności czy słowa P i Q są sobie równe czy nie

funkcja Podśłowo(słowoP , słowoQ): zwraca TAK lub NIE w zależności czy słowa P i Q są sobie równe, wykorzystując SUBSTRING(Q,P)

```
{
    if SUBSTRING(słowoP, słowoQ) = TAK and SUBSTRING(słowoQ, słowoP) = TAK then
        return TAK
    else
        return NIE
}
```

