```
zadanie algorytmiczne nr 9:
```

algorytm będzie dla zadanej listy słów L oraz słowa P przechodził po liscie sprawdzając kolejne słowa czy są równe słowu P, a jeżeli któreś bedzie równe to zwracał TAK

```
funkcja MEMBERSHIP(listaL, słowoP): zwraca TaK lub NIE w zależności czy słowoP
występuje w liście czy nie, przy wykorzystaniu EQWORD(P,Q)
{
       list L:= listaL
       if L = [-] then % sprawdzamy możliwe sprzeczności %
              return NIE
       else
                                   % jeżeli nie ma sprzeczności to zaczynamy
                                     przechodzić po głowach listy L
              if EQWORD(HEAD(L), slowoP) = TAK then
                     retrurn TAK
              else
                     while TAIL(L) != [-] do
                            L := TAIL(L)
                            if EQWORD(HEAD(L), słowoP) = TAK then
                                   return TAK
              return NIE
                            % jeżeli dotąd nie było TAk to zwracamy NIE %
}
```

zadanie algorytmiczne nr 10:

algorytm będzie sprawdzał kolejne słowa w liscie L i jeżeli nie będą one równe słowu P to dodawał je do listy pomocniczej którą na końcu zwróci

```
funkcja REMOVE(listaL, słowoP): zwraca zadaną listę słów L bez wystąpień słowa P przy
pomocy funkcji EQWORD(P,Q)
{
       list L := listaL
       list Pom := [-]
                                   % deklarujemy potrzebne zmienne%
                                   % sprawdzamy czy lista L nie jest pusta %
       if L = [-] then
              return L
       else
              if EQWORD(HEAD(L),słowoP) = NIE then %sprawdzamy czy pierwsze
                                                       słowo listy L nie jest słowem P, a
                                                       jeśli nie to dodajemy je do listy
                                                       pomocniczej %
                     MAKELIST(HEAD(L)), Pom)
              else
                     while TAIL(L) != [-] do
                                                 % sprawdzamy pozostałe słowa listy L, a
                                                 jezeli nie są slowem P to
                                                 dodajemy je do listy Pom %
                            L = TAIL(L)
                            if EQWORD(HEAD(L),slowoP) = NIE then
                                   MAKELIST(HEAD(L),Pom)
                     return Pom
                                                 % zwracamy listę pomocniczą która
                                                 zawiera wszystkie słowo z listy L
                                                 bez wystąpień słowa P %
}
```