Adam Cherek s464922

zadanie algorytmiczne nr 15:

algorytm będzie dla języków L1 i L2, przechodził po kolejnych głowach języka L2 , dodając do L1, elementy L2 których w L1 nie ma i na końcu zwracając L1 będące wówczas sumą jeżyków L1 i L2

funkcja UNION(listaL1, listaL2): zwraca język będący sumą języków L1 i L2 przy pomocy funkcji MEMBERSHIP(L,P)

```
{
        list L1:= listaL1
        list L2:= listaL2                   % deklarujemy potrzebne zmienne %

        if L1 = [-] then                    % sprawdzamy czy żaden język nie jest pusty %
                return L2
        if L2 = [-] then
                return L1

        if MEMBERSHIP(L1,HEAD(L2)) = NIE then
                MAKELIST(HEAD(L2), L1)

        while  TAIL(L2) != [-] do           % dodajemy do L1 wszystkie słowa L2, których nie
                                              ma w L1%
                L2 := Tail(L2)
                if MEMBERSHIP(L1,HEAD(L2)) = NIE then
                        MAKELIST(HEAD(L2), L1)

        return L1
}
```

\zadanie algorytmiczne nr 16:

algorytm będzie przechodził po kolejnych głowach L1 sprawdzając czy słowa te znajdują się również w L2, a jeśli tak to będzie dodawał je do listy pomocniczej POM, pod koniec zwracając tą listę będącą na końcu częścią wspólną języków L1 i L2

funkcja MEET(listaL1, listaL2): zwraca język będący częścią wspólną języków L1 i L2 przy pomocy funkcji MEMBERSHIP(L,P)

```
{
        list L1:= listaL1
        list L2:= listaL2               % deklarujemy potrzebne zmienne %
        list POM := [-]

        if L1 = [-] then                % sprawdzamy czy żaden język nie jest pusty %
                return POM
        if L2 = [-] then
                return POm

        if MEMBERSHIP(L2,HEAD(L1)) = TAK then
                MAKELIST(HEAD(L1), POM)

        while  TAIL(L1) != [-] do       % dodajemy do POM wszystkie słowa będące
                                          jednocześnie w L1 i L2 %
                L1 := Tail(L1)
                if MEMBERSHIP(L2,HEAD(L1)) = TAK then
                        MAKELIST(HEAD(L1), POM)

        return POM
}
```