

zadanie algorytmiczne nr 17:

algorytm będzie dla języków L1 i L2, przechodził po kolejnych głowach języka L1, sprawdzając czy słowa te znajdują się w języku L2, a jeśli tak, usuwał je z języka L1, zwracając na końcu różnicę języków

funkcja SETMINUS(listaL1, listaL2): zwraca język będący różnicą języków L1 i L2 przy pomocy funkcji MEMBERSHIP(L,P) oraz REMOVE(L,P)

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

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

    if MEMBERSHIP(L2,HEAD(L1)) = TAK then
        REMOVE(L1, HEAD(L1))

    while TAIL(L1) != [-] do    % usuwamy z L1 wszystkie słowa będące w L2%
        L1 := Tail(L1)
        if MEMBERSHIP(L2,HEAD(L1)) = TAK then
            REMOVE(L1, HEAD(L1))

    return L1
}
```

zadanie algorytmiczne nr 18:

algorytm będzie przechodził po kolejnych głowach L dodając do języka pomocniczego odbicie zwierciadlane tych słów, zwracając na końcu ten język pomocniczy będący odbiciem zwierciadlanym języka L

funkcja REVLANG(listaL): zwraca język będący odbiciem zwierciadlanym języka L przy pomocy funkcji REV(P)

```
{
    list L := listaL
    list Pom := [-]

    if L = [-] then
        return L

    MAKELIST(REV(HEAD(L)), Pom)

    while TAIL(L) != [-] do
        L := TAIL(L)
        MAKELIST(REV(HEAD(L)), Pom)

    return L
}
```