

zadanie algorytmiczne nr 1:

algorytm będzie dla danego słowa P w postaci listy przechodził po każdym symbolu tego słowa licząc ile symboli w tym słowie się znajduje, suma symboli czyli długość słowa zostanie na końcu zliczania zwrócona

funkcja Długość(słowo): zwraca liczbę naturalną będącą długością słowa

```
{
    int suma := 0                % inicjujemy potrzebne zmienne %
    list P := słowo
    if P = [] then                % jeżeli słowo jest puste to zwracamy sumę równą 0 %
        return suma
    else                          % jeżeli nie jest puste to liczymy symbole aż ogon
                                % będzie pusty %
        suma := suma + 1
        while TAIL(P) != [] do
            suma := suma + 1
            P := TAIL(P)
        return suma              % zwracamy sumę po zakończeniu sprawdzania
                                % ogona %
}
```

zadanie algorytmiczne nr 2:

algorytm będzie przesuwiał się po słowie licząc o ile symboli się już przesunął a gdy dojdzie do wskazanej wartości zwróci głowę listy będącą szukany symbolem

funkcja ZnajdźZnak(słowo, pozycja): zwraca znak będący symbolem na danej pozycji w słowie

```
{
    int i := 1                    % inicjujemy potrzebne zmienne %
    list P := słowo
    if pozycja = 0 then           % sprawdzamy czy pozycja jest większa od 0 %
        return null
    if pozycja > Długość(słowo) then % i nie większa niż długość słowa %
        return null

    while i < pozycja do          % przechodzimy po słowie do zadanej pozycji %
        P := TAIL(P)
        i := i + 1

    return HEAD(P)               % zwracamy głowę gdy jesteśmy na zadanej pozycji %
}
```