* Descrierea programului:

Abordare cu metoda Programării Dinamice pentru rezolvarea problemei 4, var. 3;

Complexitate – *O(n²),* unde *n* este lungimea cuvântului dat.

* Descrierea funcției recursive:
  + 1. Pentru aflarea numărului de palindromuri din cuvânt:

Se dă cuvântul *W* de lungime n.

Știm că palindromurile triviale sunt cuvintele formate dintr-o literă și cuvintele formate din două litere identice.

Observăm că dacă „atașăm” la ambele capetele ale unui palindrom aceeași literă, obținem tot un palindrom.

Se deduce următoarea relație de recurență:

Unde x și y sunt litere și w un cuvânt oarecare.

Însă, pentru a afla numărul de palindromuri din cuvântul dat, trebuie însumate toate palindromurile de lungime până la |*W* | :

Unde sunt litere din cuvântul ;

Memoizare: se păstrează într-o matrice de dimensiuni egale cu lungimea cuvântului, indexată pe rânduri după lungimea subcuvântului și pe coloane după indexul de start al subcuvântului, valoarea funcției *palindrom* apelată pentru subcuvântul respectiv.

* + 1. Pentru partiționarea șirului în număr minim de palindromuri:

Se dă cuvântul *W* de lungime n.

Știm că pentru orice cuvânt palindrom, partiționarea este trivială(partiția conține un singur element: cuvântul inițial).

Observăm că putem începe cu prima literă a cuvântului dat și, după ce generăm o soluție optimă pentru subproblema respectivă, adăugăm progresiv litere până ajungem la cuvântul inițial.

Se deduce următoarea relație de recurență:

Unde sunt litere din cuvântul .

Memoizare: se păstrează într-un vector, indexat după lungimea subcuvântului, numărul de partiții optime și poziția ultimului separator după care se află ultima partiție calculată.