Code for question 1 -

swi-prolog

```
encode([to, strike], especially).

encode(men, deans).
encode(hijacking, inspecting).
encode(commercial, departments).

encode([aircrafts], [and,faculties]).

encode([pentagon], [the, faculty, of, maths]).

encode([white, house], [faculty, of, science]).

encode([world, trade, center], [faculty, of, law]).

encode([used, as, suicide, weapons], recruited).

encode(the, the).
encode(date, date).
encode(is, is).
encode(9_11, 9_11).
encode(., .).
encode(19, 19).
encode(will, will).
encode(be, be).
encode(they, they).
encode(several, several).
encode(nineteen, nineteen).
encode(,,,).
encode(and, and).

write_list([]).
write_list([H|T]):-
   write(H),
   write(' '),
   write_list(T).

alter([], []).

%one on one
alter([H|T], [M|N]):-
   encode(H,M),
   alter(T,N).

%two on one
alter([H,Q|T], [M|N]):-
   encode([H,Q],M),
   alter(T, N).

%one on two
alter([H|T], [M,O|N]):-
   encode([H], [M,O]),
   alter(T, N).
```

```
%three on three
alter([H,I,J|T], [M,O,P|N]):-
    encode([H,I,J], [M,O,P]),
    alter(T, N).

%one on four
alter([H|T], [M,O,P,Q|N]):-
    encode([H], [M,O,P,Q]),
    alter(T, N).

%two on three
alter([H,I|T], [M,O,P|N]):-
      encode([H,I], [M,O,P]),
      alter(T,N).

%four on one
alter([H,I,J,K|T], [M|N]):-
    encode([H,I,J,K],M),
    alter(T, N).

message():-
    read(X),
    atomic_list_concat(L,' ',X),
    alter(L, V),nl,
    write('-------------------------'),nl,
    write_list(V).
```

code for question - 2

javascript code

```
var circles = [];
var noOfCircles = 0;
var circRadius  =0;
var totalInfected = 0;

function setup() {
  createCanvas(450, 450);
  background(56,193,56,50);

  while (noOfCircles < 15) {
    var circle = {
      x: random(20, width - 20),
      y: random(20, height - 20),
      r: 20,
      affected: 0
    };

    var overlapping = false;
    var d=0;
    for (var j = 0; j < circles.length; j++) {
      other = circles[j];
      d = dist(circle.x, circle.y, other.x, other.y);
      circRadius = circle.r + other.r;
      if (d < circRadius) {
        //They are overlapping!
        overlapping = true;
        break;
      }
    }

    if (!overlapping) {
      noOfCircles++;
      circles.push(circle);
    }
  }

  for (var i = 0; i < circles.length; i++) {
    fill(255, 0, 150, 100);
    noStroke();
    ellipse(circles[i].x, circles[i].y, circles[i].r * 2);
  }



  var pathogen = {
        x: random(20, width - 20),
      y: random(20, height - 20),
      r: 2
  };
  var prevX = pathogen.x
  var prevY = pathogen.y

  var abc = 0;
  while(abc<30) {
    abc++;

    fill(0,0,200,100);
    ellipse(pathogen.x, pathogen.y,pathogen.r*5);
    calculateAffected(pathogen.x, pathogen.y);
    var randX = random(-20,20)
    var randY = random(-20,20)
    pathogen.x = prevX+randX;
    pathogen.y = prevY+randY;
    prevX = pathogen.x;
    prevY = pathogen.y;

  }

  calculateNode();

  for(var q=0;q<circles.length;q++) {

    if(circles[q].affected > 0) {
      fill(0);
      ellipse(circles[q].x, circles[q].y, circles[q].r * 2);
    }
  }

  print('Total people infected : ');
  print(totalInfected);

}

function calculateNode() {
  var d = 0;
  for(var i=0;i<circles.length;i++) {
    for(var j=0;j<circles.length;j++) {
      d=dist(circles[i].x,circles[i].y,circles[j].x,circles[j].y);
      if(circles[i].affected>0) {
        if(d < 120) {
          totalInfected++;
          fill(200);
      ellipse(circles[j].x, circles[j].y, circles[j].r * 2);
        }
      }

    }
  }
}

function calculateAffected(pathogenX, pathogenY) {
  for(var i=0;i<circles.length;i++) {
    distance = dist(circles[i].x,circles[i].y,pathogenX,pathogenY);
    if(distance < 40) {
      var rand = random(0,1);
      if(rand <= 0.7) {
      circles[i].affected += 1;
      }
    }
  }

}

function draw() {

}
```