*** Two arrays must be declared:
!*** y (4) - an array containing the coordinates x = y (1), y=y(2) and projections of velocity vector
!*** Vx = y (3),Vy=y(4).
!*** The working array work (27) with a length to be calculated using the equation
!*** 3 + 6 * neqn (neqn- is number of equations)
!*** m_maa variable must also be declared as REAL. Here we take into account the Fortran
!*** feature. If variables are not declared at the beginning of the program by the REAL,
!*** INTEGER, etc. operators, the Fortran compiler can declare and create them by using the
!*** default rule: if the first character in the variable name is **i, j, k, l, m, n** then the variable must be
!*** INTEGER if it is not so then the variable must be REAL.
real y(4),mass,work(27),k
** Declaration of additional integer work array iwork(5) with fixed length 5 (defined in subroutine rkf45)
integer iwork(5)
!*** Description the global variables which must be used in subroutine "func" to calculate
!*** derivatives of coordinate x,y and velocity projections Vx,Vy with respect to time
common param,r0
!*** external operator should be used to describe the variable “func” as a name of external subroutine
external func
!*** Opening files to reading and saving data
open(10,file="ruther.dat")
!*** Some additional parameters (descriptions you can find in LOENG_28.04.2020)
b=0.01
r0=1.
q1=3.e-6
q2=0.1e-6
k=9.e9
mass=1.e-3
param=q1*q2*k/mass
!*** Absolute and realative errors
relerr=1.e-5
abserr=1.e-5
!*** timestep
dt=0.0001
!*** number of timesteps
nt=100
!*** initializing of iflag and time parameters
iflag=1
t=0.
!*** number of equations
neqn=4
!*** Initial conditions
y(1)=0.
y(2)=b
y(3)=10.
y(4)=0.
!*** Cycle operator to perform the integration of the system of differential equations and
!*** calculation of coordinates and velocities at different time moments with timestep dt
!*** The total time of simulation can be calculated as a product of variables nt and dt

```
do it=1,nt
!*** Calculation of new value of time (should be done by hand)
tout=t+dt
*** Calculation of coordinate y(1),y(2)and velocities y(3),y(4) at the next time moment tout=t+dt
call rkf45(func,neqn,y,t,tout,relerr,abserr,iflag,work,iwork)
!*** Just in case the control of the iflag value.  If iflag=2 the calculation was success and we can to continue
if(iflag.ne.2) then
if (iflag.ne.2) then
iflag=2
endif
!***Saving data in the following format
!*** time x y Vx Vy
write(10,*) tout,y
enddo
!*** stop program
stop
!*** end of source code
end
!*** The most important part of the program. Here we must implement the differential equation.
!*** Now we  need to calculate the derivatives with respect to the coordinate and velocity
!*** with respect to time.
!*** Now the subroutine ″func″ must be created. Number of input parameters is fixed.
!*** t-time, y(6)-array consist the coordinate y(1),y(2) and velocity projections  y(4),y(5) of particle,
!*** yp(4)-array with derivatives so that:
!*** dy(1)=dy(1)/dt=velocity_x=y(3)
!*** dy(2)=dy(2)/dt=velocity_y=y(4)
!*** dy(3)=param*(y(1)-r0)/(sqrt((y(1)-r0)**2+y(2)**2))**3
!*** dy(4)=param*y(2)/(sqrt((y(1)-r0)**2+y(2)**2))**3
subroutine func(t,y,yp)
real y(4),yp(4)
common param,r0
yp(1)=y(3)
yp(2)=y(4)
yp(3)=param*(y(1)-r0)/(sqrt((y(1)-r0)**2+y(2)**2))**3
yp(4)=param*y(2)/(sqrt((y(1)-r0)**2+y(2)**2))**3
return
end
```