*** Two arrays must be declared:
!*** y (6) - an array containing the coordinates x = y (1), y=y(2),z=y(3) and projections of velocity vector
!*** Vx = y (4),Vy=y(5),Vz=y(6).
!*** The working array work (39) 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.
!*** here b(3) - is the magnetic induction vector, e (3) - is the electric field strength vector
real y(6),work(39),m
!*** 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 velocities projections Vx,Vy with respect to time
common qm,r_maa
!*** external operator should be used to describe the variable “func” as a name of external subroutine
external f
!*** Opening files to reading and saving data
open(20,file="lorentz_prooton.dat")
!*** Some additional parameters
!*** Osakese laeng ja mass
q=1.602e-19
m=1.67e-27
!*** Maa raadius
r_maa=6378.
!*** Aja sammude arv
nt=20000
!*** Ajasamm
dt=0.01
!*** Absoluutne ja suhteline viga
realerr=1.e-6
abserr=1.e-6
!*** Algtingimused
!*** Prootoni algus koordinaadid x=38000km, y=z=0 km
y(1)=38000.
y(2)=0.
y(3)=0.
*** Prootonite esialgse kiirusvektori projektsioonid V_x=0 km/s, V_y=10000 km/s, V_z=20000 km/s
y(4)=0.
y(5)=1.e4
y(6)=2.e4

qm=q/m
!*** võrrandite arv neqn=6
neqn=6
!*** Initializing of the time and iflag

```
t=0
iflag=1
!*** 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 i=1,nt
!*** Calculation of new value of time (should be done by hand)
tout=t+dt
*** Calculation of coordinate y(1),y(2),y(3) and velocitie y(4),y(5),y(6) at the next time moment tout=t+dt
call rkf45(f,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
iflag=2
endif
!***Saving data in the following format
!*** x y z
write(20,*) y(1:3)
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),y(3) and velocity projections  y(4),y(5),y(6) of sphere,
!*** yp(6)-array with derivatives so that:
!*** dy(1)=dy(1)/dt=velocity_x=y(4)
!*** dy(2)=dy(2)/dt=velocity_y=y(5)
!*** dy(3)=dy(3)/dt=velocity_z=y(6)
!*** dy(4)=qm*(y(5)*b(3)-y(6)*b(2))
!*** dy(5)=qm*(y(6)*b(1)-y(4)*b(3))
!*** dy(6)=qm*(y(4)*b(2)-y(5)*b(1))
subroutine f(t,y,dy)
!*** Declaration of arrays
real y(6),dy(6),b(3),re(3)
!*** Description the global variables which must be used in subroutine "func" to calculate
!*** derivatives of coordinate x and velocities projections  Vx with respect to time
common qm,r_maa
!*** Calculation of dimensionless coordinates of proton
re(1)=y(1)/r_maa
re(2)=y(2)/r_maa
re(3)=y(3)/r_maa
!*** Calculation of induction vector
call emag(re,b)
```

```
! *** Calculation of first derivative for coordinates  with respect to time
dy(1)=y(4)
dy(2)=y(5)
dy(3)=y(6)
dy(4)=qm*(y(5)*b(3)-y(6)*b(2))
dy(5)=qm*(y(6)*b(1)-y(4)*b(3))
dy(6)=qm*(y(4)*b(2)-y(5)*b(1))
return
end

!*** Subroutine to calculate the Earth's magnetic field (induction vector)
!*** here: r (3)- contains the coordinates of the point at which the induction vector should be calculated,
!*** and b(3) – calculated projections of induction vector b(1)=Bx,  b(2)=By, b(3)=Bz (see lecture)
subroutine emag(r,b)
real r(3), b(3)
b0=3.07e-05
x=r(1)
y=r(2)
z=r(3)
rr=sqrt(x**2+y**2+z**2)
b(1)=-3.*b0*z*x/(rr**5)
b(2)=-3.*b0*z*y/(rr**5)
b(3)=-b0*(2*z**2-x**2-y**2)/(rr**5)
return
end
```