```
*** 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 work(27),y(4),m_maa
!*** 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 gm
!*** external operator should be used to describe the variable “func” as a name of external subroutine
external func
!*** Now we need to open the file to save the calculated results
open(10,file=″kepler.dat″)
!*** Relative and absolute errors for calculation the coordinate and velocity this is a input parameter for
!*** subroutine rkf45
relerr=1.e-7
abserr=1.e-7
!*** On the next lines some important variables should be created and we must assign
!*** values for them
!*** Mass of he Earth in kilograms
m_maa=5.9722e+24
!*** Radius of the Earth in kilometers
r_maa=6378.
!*** Gravity constant
G=6.67408e-11
!*** Additional work parameter  (the number 1.e-9 arose due to conversion of meters into kilometers)
gm=G*m_maa*1.e-9
!*** Number of differential equations
neqn=4
!*** Initial conditions
!*** y(1)=6578 km       – x coordinate of sphere at start time moment
!*** y(2)=0 km          – y coordinate of sphere at start time moment
!*** y(3)=0. km/s       – velocity (x-projection) of sphere at start time moment
!*** y(4)=9.78337 km/s- velocity (y-projection)  of sphere at start time moment
y(1)=6578.
y(2)=0.
y(3)=0.
```

```
y(4)=9.78337
!*** Total number of time steps
nt=2000
!*** Time step in seconds
dt=10.
!*** Initial value of time
t=0.
!*** iflag=1 is need to start the calculation, it means  that we are starting a new simulation
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=0.1 seconds.
!*** The total time of simulation can be calculated as a product of variables  nt and dt ,
!*** in our  case it is equal to nt*dt=10*200=2000 seconds.
do i=1,nt
!*** Caslculation of new value of time (should be done by hand)
tout=t+dt
!*** Calculation of coordinates y(1),y(2) and velocity projections 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  continue
if(iflag.ne.2)then
iflag=2
endif
!*** Calculation of the absolute value of velocity ( |v|=sqrt(v_x^2+v_y^2) )
vv=sqrt(y(3)**2+y(4)**2)
!*** Calculation of the distance between satellite and center of the Earth ( |r|=sqrt(x^2+y^2) )
rr=sqrt(y(1)**2+y(2)**2)
!*** Calculation of the projections and modulus of acceleration vector
ax=-gm*y(1)/sqrt(y(1)**2+y(2)**2)**3
ay=-gm*y(2)/sqrt(y(1)**2+y(2)**2)**3
aa=sqrt(ax**2+ay**2)
!*** Saving data in tne next format
!*** time  x y Vx Vy acceleration
!*** only 6 parameters
write(10,*) tout,y,rr,vv,aa
!*** End of cycle operators
enddo

!*** Stop the calculations
stop
!*** End of program
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(4)-array consist the coordinate y(1),y(2) and velocity projections y(3),y(4) of sphere,
!*** yp(4)-array with derivatives so that:
!*** dy(1)=dy(1)/dt=velocity=y(3) and
!*** dy(2)=dy(2)/dt=velocity=y(4) and
!*** dy(3)=dy(2)/dt=acceleration=-gm*y(1)/sqrt(y(1)**2+y(2)**2)**3
!*** dy(4)=dy(2)/dt=acceleration=-gm*y(2)/sqrt(y(1)**2+y(2)**2)**3
!*** (theoretical background you can find in precis of lecture)
subroutine func(t,y,dy)
!*** 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 gm
!*** Declaration of arrays
real y(4),dy(4)
!*** first derivative for coordinate x with respect to time
dy(1)=y(3)
!*** first derivative for coordinate y with respect to time
dy(2)=y(4)
!*** Second derivative, calculation of x-projection of acceleration with the second Newton′s low
dy(3)=-gm*y(1)/sqrt(y(1)**2+y(2)**2)**3
!*** Second derivative, calculation of y-projection of acceleration with the second Newton′s low
dy(4)=-gm*y(2)/sqrt(y(1)**2+y(2)**2)**3
!*** Back to call operator in the main program
return
!*** End of source code for subroutine "func"
end
```