```
program Solar_elevation_angle

use Declination_angle
use Solar_hour_angle

implicit none
real                          :: pi
real :: d, h, SEA, lat, Declination, localsolartime, angle
integer,dimension(12) :: months_num
integer :: mon,day,year

pi            =    3.14
months_num          = (/31,28,31,30,31,30,31,31,30,31,30,31/)

year = 2021
mon    = 12
day    = 31

d = 365-sum(months_num(mon:))+day

localsolartime = 10.53

Declination=d+localsolartime/24
call Declination_angle(Declination,angle)
call solar_hour_angle(localsolartime,h)

if (A < 0) then
       A=-A
  endif

lat = 22.54

SEA                                         =                                         180-
asin(sin(lat*pi/360)*sin(A*pi/360)+cos(lat*pi/360)*cos(A*pi/360)*cos(h*pi/360))*360/pi


endprogram Solar_elevation_angle




module Declination_angle
```

```
implicit none

contains

subroutine Declination_angle(Declination,angle)
	implicit none
	real, intent(in)   :: Declination
	real, intent(out)      :: angle
	real                   :: pi

	pi = 3.1415926

	angle = asin(sin(-23.44*pi/180)*cos((360/365.24*(D+10)+360*0.0167*sin(360*(D-2)/365.24)/pi) * pi / 180.))* 180. / pi


end subroutine Declination_angle

end module Declination_angle




module Solar_hour_angle
implicit none

contains

subroutine Solar_hour_angle( localsolartime ,h)
	real, intent(in) :: localsolartime
	real, intent(out)   :: h

	h = 15*( localsolartime -12)


end subroutine Solar_hour_angle

end module Solar_hour_angle
```