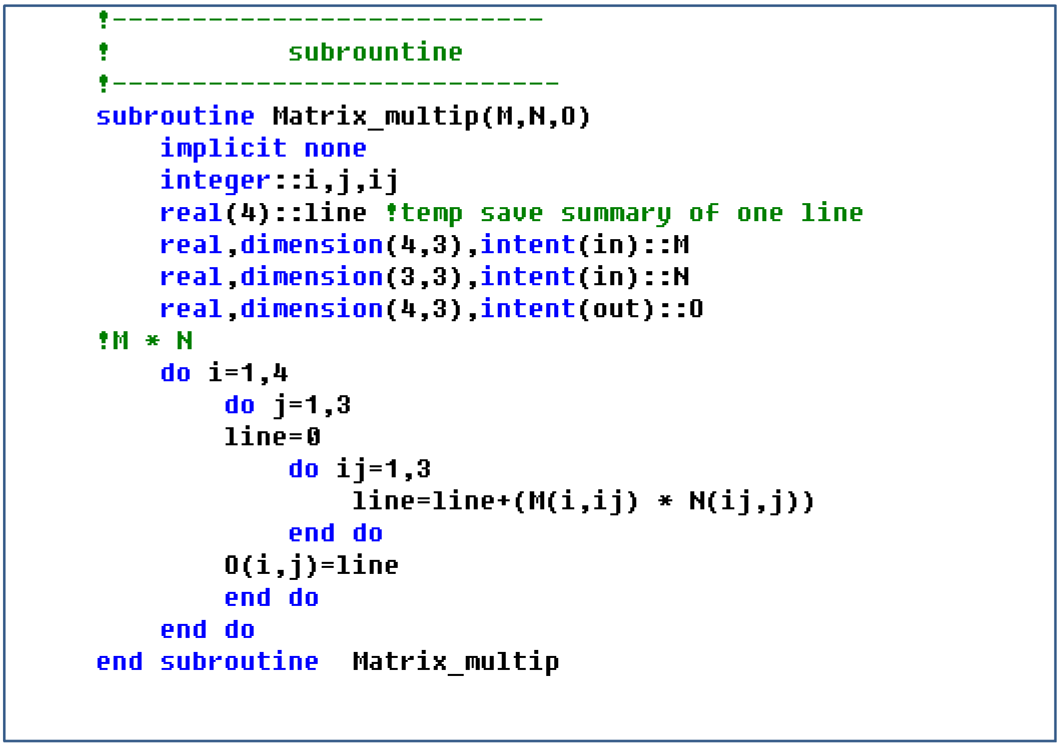
Assignment 06

1. Matrix multiplication

1.1 **[5 points]** Write a subroutine *Matrix_multip.f90* to do matrix multiplication.

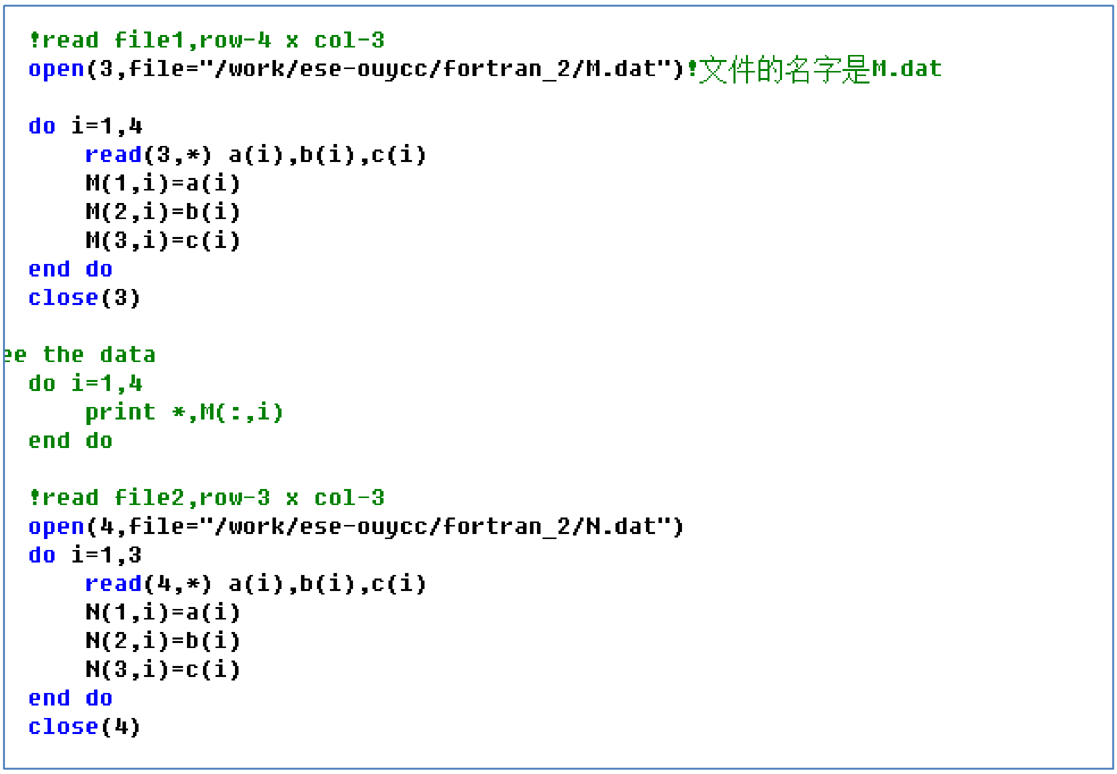


```
!---------------------------
!           subrountine
!----------------------------
subroutine Matrix_multip(M,N,O)
    implicit none
    integer::i,j,ij
    real(4)::line !temp save summary of one line
    real,dimension(4,3),intent(in)::M
    real,dimension(3,3),intent(in)::N
    real,dimension(4,3),intent(out)::O
!M * N
    do i=1,4
        do j=1,3
        line=0
            do ij=1,3
                line=line+(M(i,ij) * N(ij,j))
            end do
        O(i,j)=line
        end do
    end do
end subroutine  Matrix_multip
```

1.2 [5 points] Write a program *Main.f90* to

read */work/ese-ouycc/fortran_2/M.dat* as the matrix *M*,

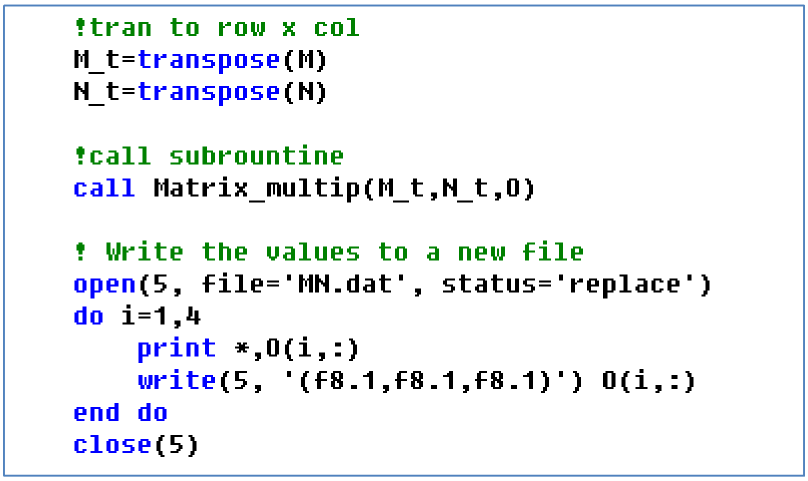
and */work/ese-ouycc/fortran_2/N.dat* as the matrix *N*.



```
  !read file1,row-4 x col-3
  open(3,file="/work/ese-ouycc/fortran_2/M.dat")!文件的名字是M.dat

  do i=1,4
      read(3,*) a(i),b(i),c(i)
      M(1,i)=a(i)
      M(2,i)=b(i)
      M(3,i)=c(i)
  end do
  close(3)

ee the data
  do i=1,4
      print *,M(:,i)
  end do

  !read file2,row-3 x col-3
  open(4,file="/work/ese-ouycc/fortran_2/N.dat")
  do i=1,3
      read(4,*) a(i),b(i),c(i)
      N(1,i)=a(i)
      N(2,i)=b(i)
      N(3,i)=c(i)
  end do
  close(4)
```

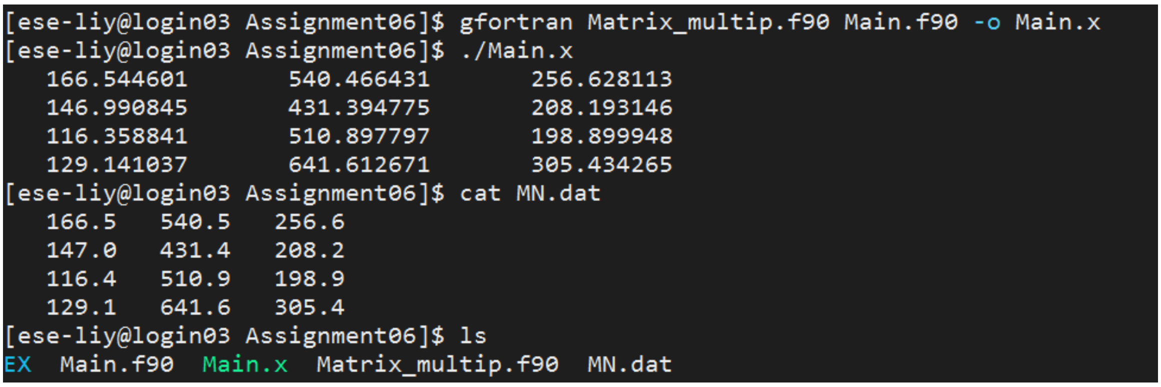
1.2 **[5 points]** Call subroutine *Matrix_multip()* from *Main.f90* to compute *M*N*; write the output to a new file *MN.dat*, values are in formats of *f8.1*.

#Call subroutine:



```
!tran to row x col
M_t=transpose(M)
N_t=transpose(N)

!call subrountine
call Matrix_multip(M_t,N_t,O)

! Write the values to a new file
open(5, file='MN.dat', status='replace')
do i=1,4
    print *,O(i,:)
    write(5, '(f8.1,f8.1,f8.1)') O(i,:)
end do
close(5)
```

```
#execute the program and view the output file
```



```
[ese-liy@login03 Assignment06]$ gfortran Matrix_multip.f90 Main.f90 -o Main.x
[ese-liy@login03 Assignment06]$ ./Main.x
   166.544601       540.466431       256.628113
   146.990845       431.394775       208.193146
   116.358841       510.897797       198.899948
   129.141037       641.612671       305.434265
[ese-liy@login03 Assignment06]$ cat MN.dat
   166.5   540.5   256.6
   147.0   431.4   208.2
   116.4   510.9   198.9
   129.1   641.6   305.4
[ese-liy@login03 Assignment06]$ ls
EX  Main.f90  Main.x  Matrix_multip.f90  MN.dat
```

2.Calculate the solar zenith angle

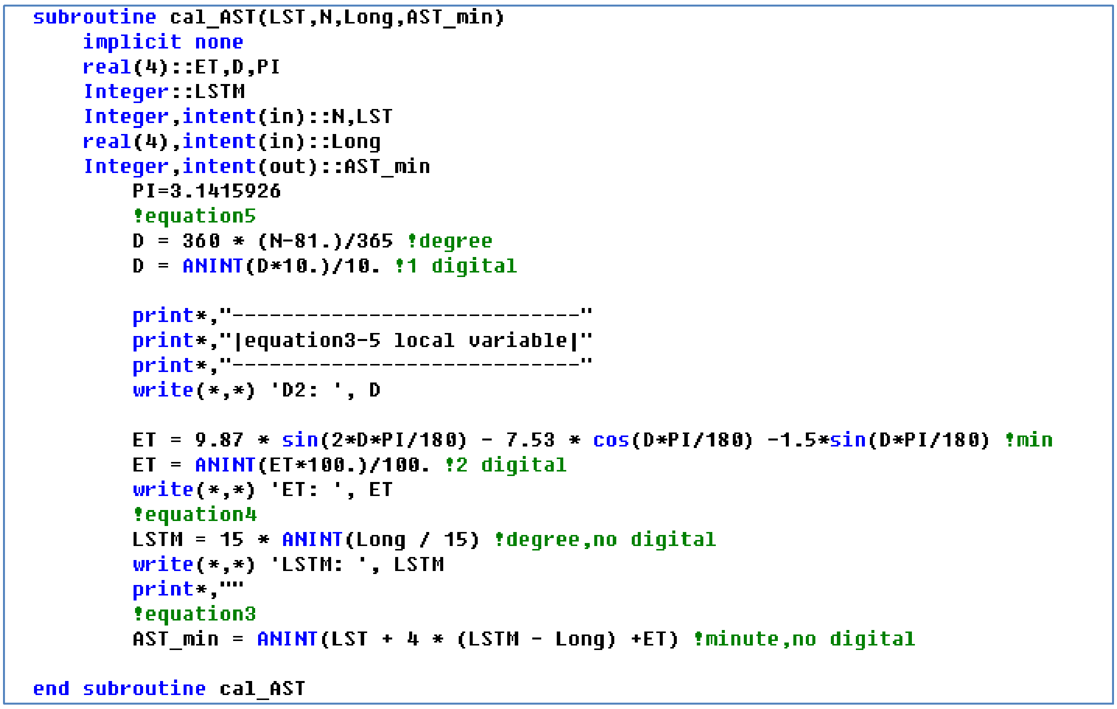
2.1 [5 points] Write a module *Declination_angle* to calculate the declination angle on a certain date.

[**Hint:** using equation 2]

```
module Declination_angle
!implicit none

contains
        subroutine cal_dec_angle(N,delta)
            implicit none
            real(4)::PI
            Integer,intent(in)::N
        real(4),intent(out)::delta
            PI=3.1415926
                !equation2
                delta= 23.45 * SIN((N+284.)/365 * 360*PI/180)

        end subroutine cal_dec_angle

end module Declination_angle
```

2.2 [10 points] Write a module *AST* to calculate the apparent solar time (AST; or local solar time) in a certain location for a certain date and time.

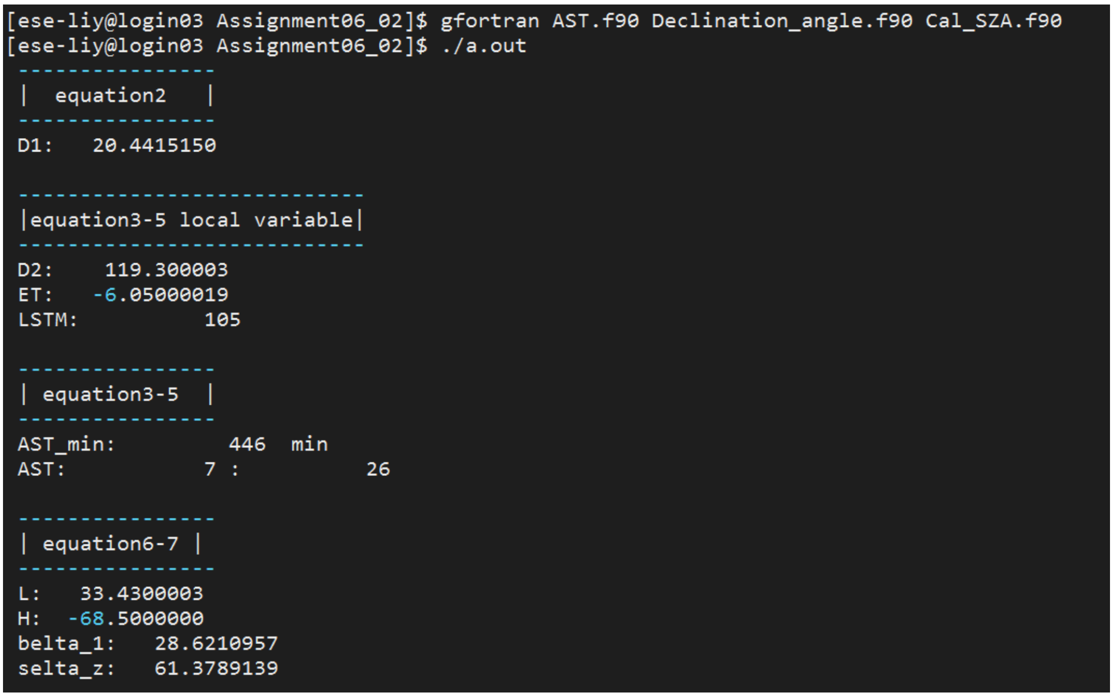
[**Hint:** using equation 3-5]



```
subroutine cal_AST(LST,N,Long,AST_min)
    implicit none
    real(4)::ET,D,PI
    Integer::LSTM
    Integer,intent(in)::N,LST
    real(4),intent(in)::Long
    Integer,intent(out)::AST_min
        PI=3.1415926
        !equation5
        D = 360 * (N-81.)/365 !degree
        D = ANINT(D*10.)/10. !1 digital

        print*,"----------------------------"
        print*,"|equation3-5 local variable|"
        print*,"----------------------------"
        write(*,*) 'D2: ', D

        ET = 9.87 * sin(2*D*PI/180) - 7.53 * cos(D*PI/180) -1.5*sin(D*PI/180) !min
        ET = ANINT(ET*100.)/100. !2 digital
        write(*,*) 'ET: ', ET
        !equation4
        LSTM = 15 * ANINT(Long / 15) !degree,no digital
        write(*,*) 'LSTM: ', LSTM
        print*,""
        !equation3
        AST_min = ANINT(LST + 4 * (LSTM - Long) +ET) !minute,no digital

end subroutine cal_AST
```

2.3 [10 points] Write a main program (*Cal_SZA.f90*) that uses module *Declination_angle* and *AST* to print the SZA in a certain location for a certain date and time.

[**Hint:** using equation 6-7]

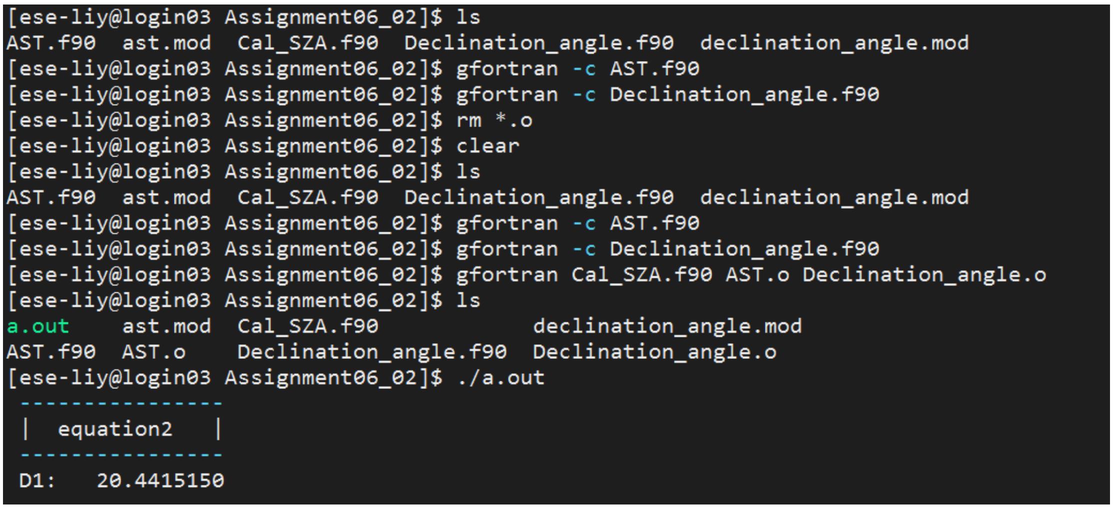


```
[ese-liy@login03 Assignment06_02]$ gfortran AST.f90 Declination_angle.f90 Cal_SZA.f90
[ese-liy@login03 Assignment06_02]$ ./a.out
 ----------------
 |  equation2   |
 ----------------
 D1:   20.4415150

 ----------------------------
 |equation3-5 local variable|
 ----------------------------
 D2:    119.300003
 ET:   -6.05000019
 LSTM:         105

 ----------------
 | equation3-5  |
 ----------------
 AST_min:         446  min
 AST:           7 :          26

 ----------------
 | equation6-7 |
 ----------------
 L:   33.4300003
 H:  -68.5000000
 belta_1:   28.6210957
 selta_z:   61.3789139
```

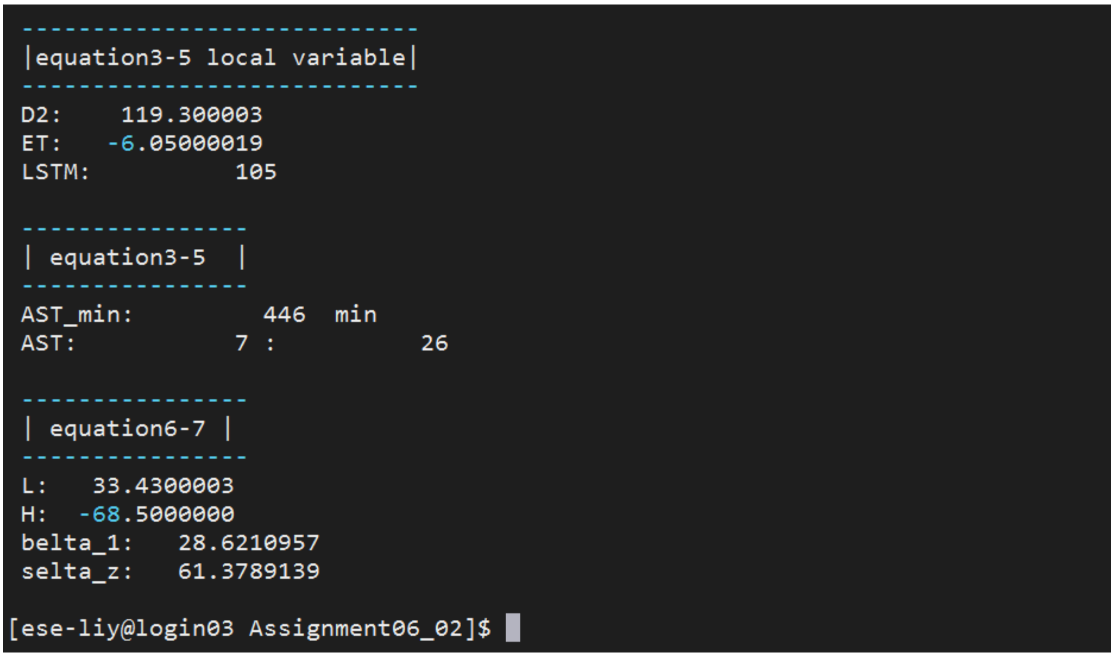
2.4 [5 points] Create a library (*libsolar.a*) that

contains *Declination_angle.o* and *AST.o*. Compile *Cal_SZA.f90* using *libsolar.a*.

```
#execute the program with .o file
```

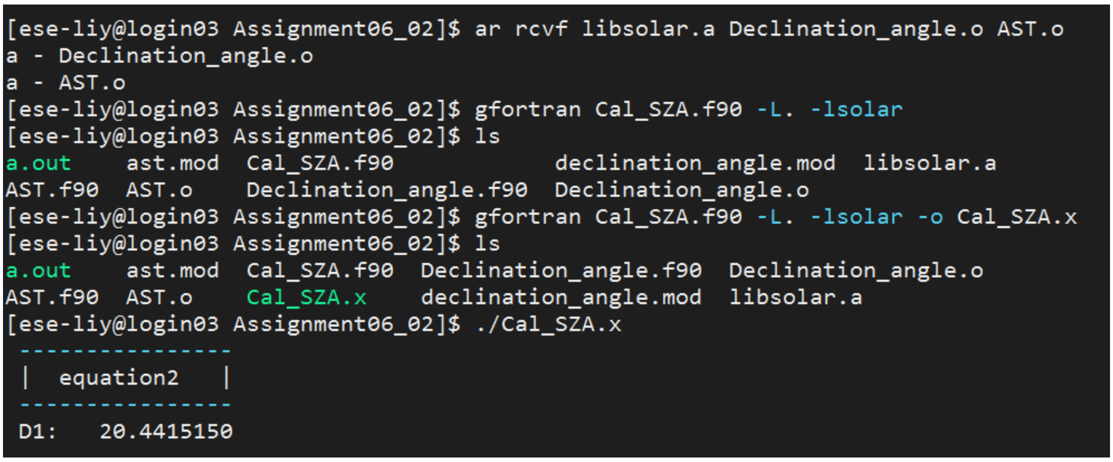


```
[ese-liy@login03 Assignment06_02]$ ls
AST.f90  ast.mod  Cal_SZA.f90  Declination_angle.f90  declination_angle.mod
[ese-liy@login03 Assignment06_02]$ gfortran -c AST.f90
[ese-liy@login03 Assignment06_02]$ gfortran -c Declination_angle.f90
[ese-liy@login03 Assignment06_02]$ rm *.o
[ese-liy@login03 Assignment06_02]$ clear
[ese-liy@login03 Assignment06_02]$ ls
AST.f90  ast.mod  Cal_SZA.f90  Declination_angle.f90  declination_angle.mod
[ese-liy@login03 Assignment06_02]$ gfortran -c AST.f90
[ese-liy@login03 Assignment06_02]$ gfortran -c Declination_angle.f90
[ese-liy@login03 Assignment06_02]$ gfortran Cal_SZA.f90 AST.o Declination_angle.o
[ese-liy@login03 Assignment06_02]$ ls
a.out    ast.mod  Cal_SZA.f90            declination_angle.mod
AST.f90  AST.o    Declination_angle.f90  Declination_angle.o
[ese-liy@login03 Assignment06_02]$ ./a.out
 ----------------
 |  equation2   |
 ----------------
 D1:   20.4415150
```



```
 ------------------------------
 |equation3-5 local variable|
 ------------------------------
 D2:    119.300003
 ET:   -6.05000019
 LSTM:         105

 ----------------
 | equation3-5  |
 ----------------
 AST_min:         446  min
 AST:           7 :          26

 ----------------
 | equation6-7 |
 ----------------
 L:   33.4300003
 H:  -68.5000000
 belta_1:   28.6210957
 selta_z:   61.3789139

[ese-liy@login03 Assignment06_02]$ 
```

#execute the program with library



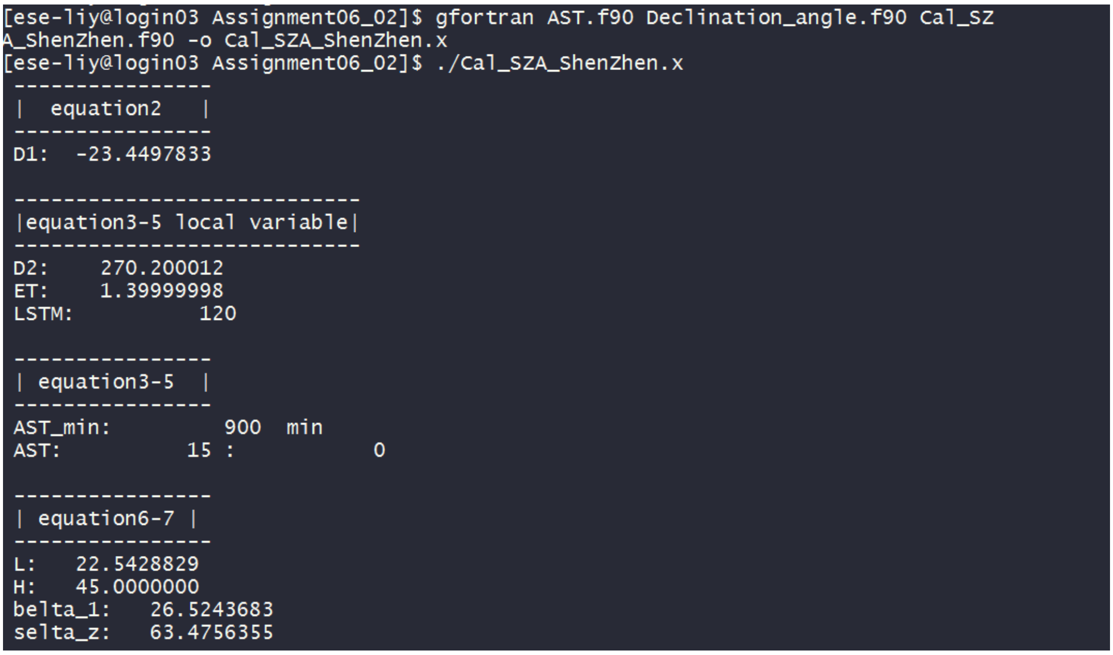
```
[ese-liy@login03 Assignment06_02]$ ar rcvf libsolar.a Declination_angle.o AST.o
a - Declination_angle.o
a - AST.o
[ese-liy@login03 Assignment06_02]$ gfortran Cal_SZA.f90 -L. -lsolar
[ese-liy@login03 Assignment06_02]$ ls
a.out    ast.mod  Cal_SZA.f90            declination_angle.mod  libsolar.a
AST.f90  AST.o    Declination_angle.f90  Declination_angle.o
[ese-liy@login03 Assignment06_02]$ gfortran Cal_SZA.f90 -L. -lsolar -o Cal_SZA.x
[ese-liy@login03 Assignment06_02]$ ls
a.out    ast.mod  Cal_SZA.f90  Declination_angle.f90  Declination_angle.o
AST.f90  AST.o    Cal_SZA.x    declination_angle.mod  libsolar.a
[ese-liy@login03 Assignment06_02]$ ./Cal_SZA.x
 ----------------
 |  equation2   |
 ----------------
 D1:   20.4415150
```

So the library is loaded and ran well.

2.5 [5 points] Print the SZA for Shenzhen (*22.542883N, 114.062996E*)

at *14:35* (Beijing time; UTC+8) on *2020-12-20*.

#work flow of the script wrote in Fortran



```
[ese-liy@login03 Assignment06_02]$ gfortran AST.f90 Declination_angle.f90 Cal_SZ
A_ShenZhen.f90 -o Cal_SZA_ShenZhen.x
[ese-liy@login03 Assignment06_02]$ ./Cal_SZA_ShenZhen.x
 ----------------
 |  equation2   |
 ----------------
 D1:  -23.4497833

 ----------------------------
 |equation3-5 local variable|
 ----------------------------
 D2:    270.200012
 ET:    1.39999998
 LSTM:          120

 ----------------
 | equation3-5  |
 ----------------
 AST_min:         900  min
 AST:          15 :          0

 ----------------
 | equation6-7 |
 ----------------
 L:   22.5428829
 H:   45.0000000
 belta_1:   26.5243683
 selta_z:   63.4756355
```

SZA; θz = 63.48