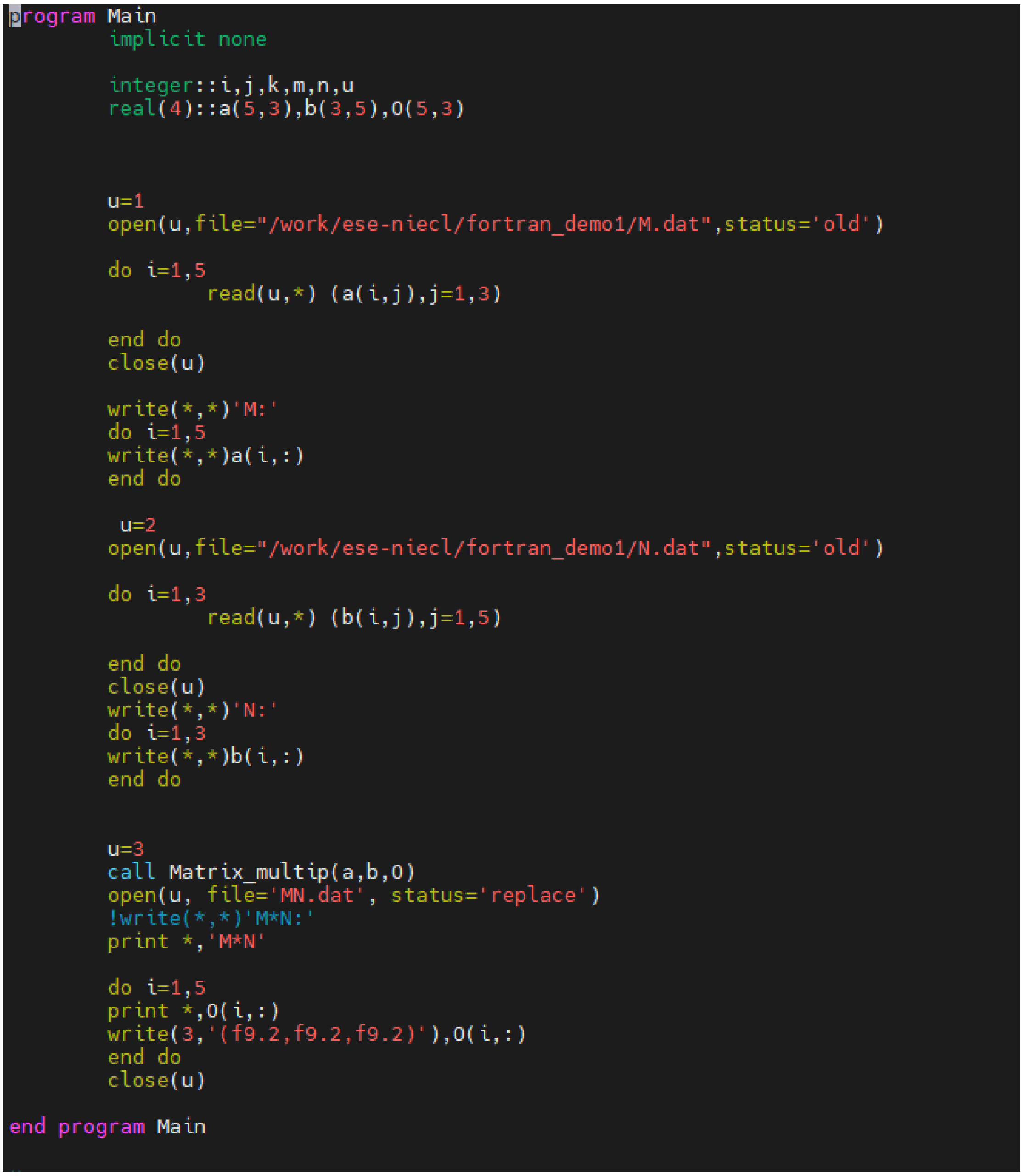
学生：聂采龙
批改：李娟
得分：39/40

1.1

Main.f90

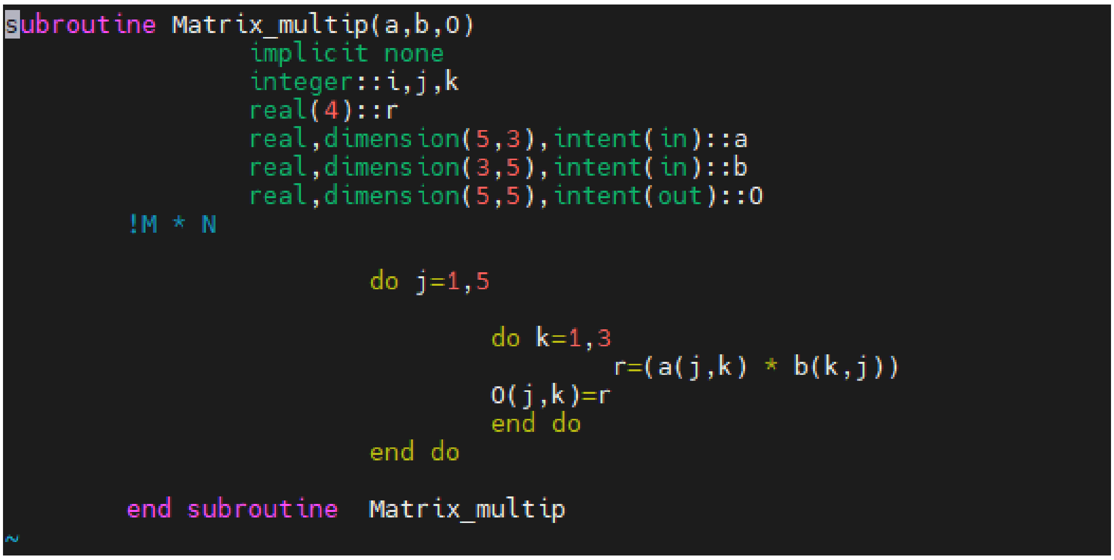
读取矩阵并打印



```
program Main
        implicit none

        integer::i,j,k,m,n,u
        real(4)::a(5,3),b(3,5),O(5,3)



        u=1
        open(u,file="/work/ese-niecl/fortran_demo1/M.dat",status='old')

        do i=1,5
                read(u,*) (a(i,j),j=1,3)

        end do
        close(u)

        write(*,*)'M:'
        do i=1,5
        write(*,*)a(i,:)
        end do

         u=2
        open(u,file="/work/ese-niecl/fortran_demo1/N.dat",status='old')

        do i=1,3
                read(u,*) (b(i,j),j=1,5)

        end do
        close(u)
        write(*,*)'N:'
        do i=1,3
        write(*,*)b(i,:)
        end do


        u=3
        call Matrix_multip(a,b,O)
        open(u, file='MN.dat', status='replace')
        !write(*,*)'M*N:'
        print *,'M*N'

        do i=1,5
        print *,O(i,:)
        write(3,'(f9.2,f9.2,f9.2)'),O(i,:)
        end do
        close(u)

end program Main
```

1.2

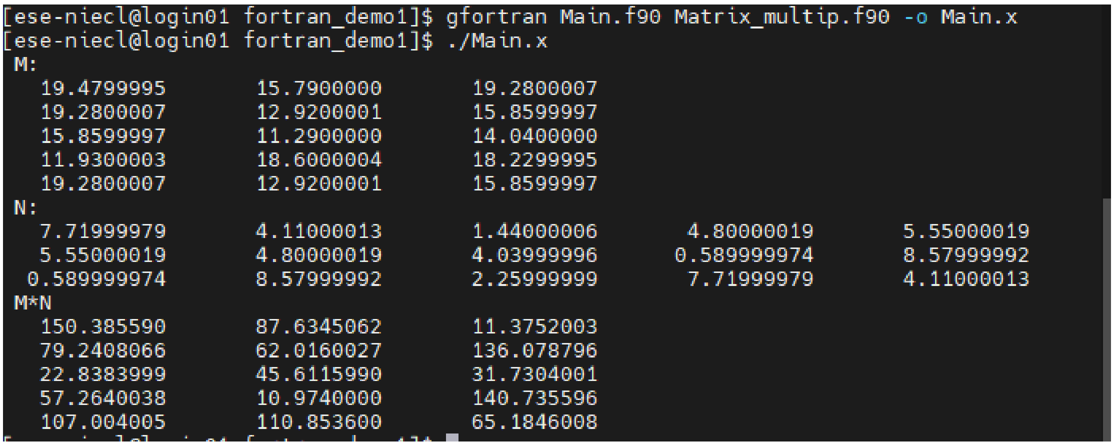
MaMatrix_multip.f90
通过循环的方式完成矩阵相乘



```
subroutine Matrix_multip(a,b,O)
                implicit none
                integer::i,j,k
                real(4)::r
                real,dimension(5,3),intent(in)::a
                real,dimension(3,5),intent(in)::b
                real,dimension(5,5),intent(out)::O
        !M * N

                        do j=1,5

                                do k=1,3
                                        r=(a(j,k) * b(k,j))
                                O(j,k)=r
                                end do
                        end do

        end subroutine  Matrix_multip
~
```

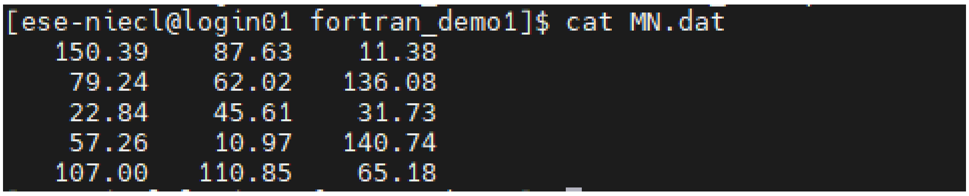
1.3

矩阵计算过程及结果
[ese-niecl@login01 fortran_demo1]$ gfortran Main.f90 Matrix_multip.f90 -o Main.x
[ese-niecl@login01 fortran_demo1]$ ./Main.x



```
[ese-niecl@login01 fortran_demo1]$ gfortran Main.f90 Matrix_multip.f90 -o Main.x
[ese-niecl@login01 fortran_demo1]$ ./Main.x
 M:
   19.4799995       15.7900000       19.2800007
   19.2800007       12.9200001       15.8599997
   15.8599997       11.2900000       14.0400000
   11.9300003       18.6000004       18.2299995
   19.2800007       12.9200001       15.8599997
 N:
   7.71999979       4.11000013       1.44000006       4.80000019       5.55000019
   5.55000019       4.80000019       4.03999996      0.589999974       8.57999992
  0.589999974       8.57999992       2.25999999       7.71999979       4.11000013
 M*N
   150.385590       87.6345062       11.3752003
   79.2408066       62.0160027       136.078796
   22.8383999       45.6115990       31.7304001
   57.2640038       10.9740000       140.735596
   107.004005       110.853600       65.1846008
```

MN.dat

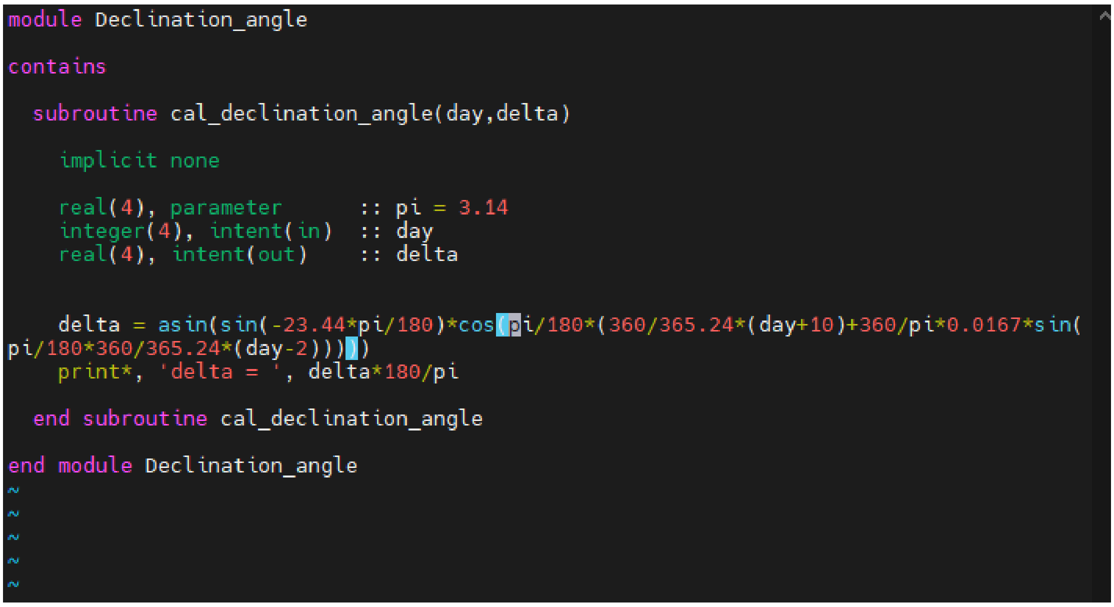


```
[ese-niecl@login01 fortran_demo1]$ cat MN.dat
   150.39     87.63     11.38
    79.24     62.02    136.08
    22.84     45.61     31.73
    57.26     10.97    140.74
   107.00    110.85     65.18
```

the result was wrong,-1

2.1

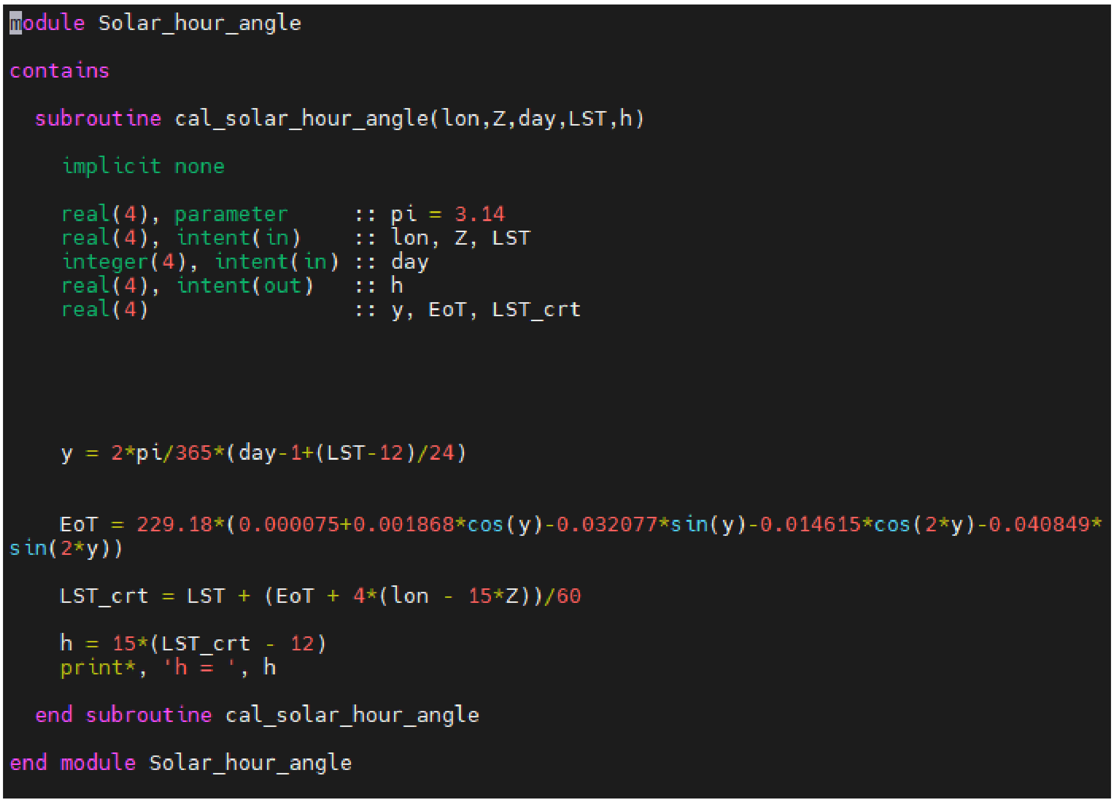
Declination_angle.f90



```
module Declination_angle

contains

  subroutine cal_declination_angle(day,delta)

    implicit none

    real(4), parameter        :: pi = 3.14
    integer(4), intent(in)   :: day
    real(4), intent(out)     :: delta


    delta = asin(sin(-23.44*pi/180)*cos(pi/180*(360/365.24*(day+10)+360/pi*0.0167*sin(
pi/180*360/365.24*(day-2)))))
    print*, 'delta = ', delta*180/pi

  end subroutine cal_declination_angle

end module Declination_angle
~
~
~
~
~
```

2.2

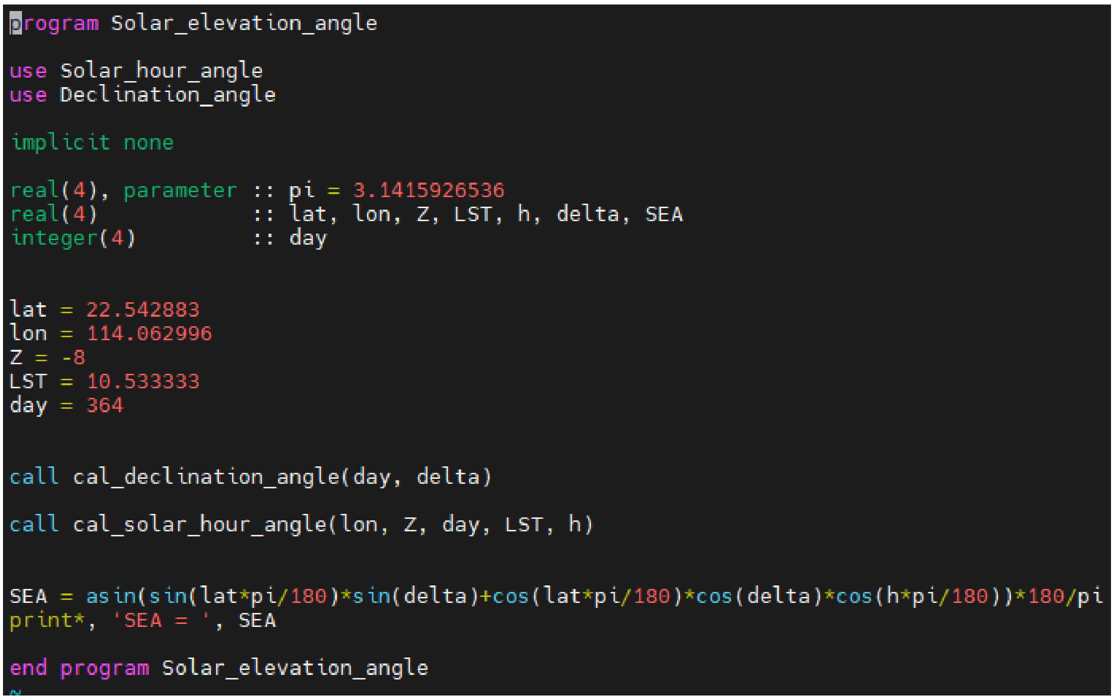
Solar_hour_angle.f90



```
module Solar_hour_angle

contains

  subroutine cal_solar_hour_angle(lon,Z,day,LST,h)

    implicit none

    real(4), parameter     :: pi = 3.14
    real(4), intent(in)    :: lon, Z, LST
    integer(4), intent(in) :: day
    real(4), intent(out)   :: h
    real(4)                :: y, EoT, LST_crt




    y = 2*pi/365*(day-1+(LST-12)/24)


    EoT = 229.18*(0.000075+0.001868*cos(y)-0.032077*sin(y)-0.014615*cos(2*y)-0.040849*
sin(2*y))

    LST_crt = LST + (EoT + 4*(lon - 15*Z))/60

    h = 15*(LST_crt - 12)
    print*, 'h = ', h

  end subroutine cal_solar_hour_angle

end module Solar_hour_angle
```

2.3

Solar_elevation_angle.f90
提前输入深圳的经纬度(22.542883N, 114.062996E) ，10:32 为10.533333。
从 UTC+8 得 Z=8 ，从日期 2021-12-31 得到 day=364

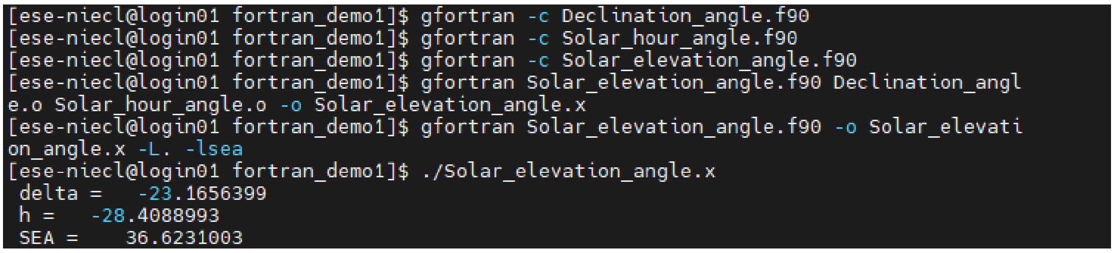


```
program Solar_elevation_angle

use Solar_hour_angle
use Declination_angle

implicit none

real(4), parameter :: pi = 3.1415926536
real(4)            :: lat, lon, Z, LST, h, delta, SEA
integer(4)         :: day


lat = 22.542883
lon = 114.062996
Z = -8
LST = 10.533333
day = 364


call cal_declination_angle(day, delta)

call cal_solar_hour_angle(lon, Z, day, LST, h)


SEA = asin(sin(lat*pi/180)*sin(delta)+cos(lat*pi/180)*cos(delta)*cos(h*pi/180))*180/pi
print*, 'SEA = ', SEA

end program Solar_elevation_angle
```

2.4

```
[ese-niecl@login01 fortran_demo1]$ gfortran -c Declination_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran -c Solar_hour_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran -c Solar_elevation_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran Solar_elevation_angle.f90 Declination_angle.o
Solar_hour_angle.o -o Solar_elevation_angle.x
[ese-niecl@login01 fortran_demo1]$ gfortran Solar_elevation_angle.f90 -o
Solar_elevation_angle.x -L. -lsea
[ese-niecl@login01 fortran_demo1]$ ./Solar_elevation_angle.x
```

构建 library，输出太阳高度角



```
[ese-niecl@login01 fortran_demo1]$ gfortran -c Declination_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran -c Solar_hour_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran -c Solar_elevation_angle.f90
[ese-niecl@login01 fortran_demo1]$ gfortran Solar_elevation_angle.f90 Declination_angl
e.o Solar_hour_angle.o -o Solar_elevation_angle.x
[ese-niecl@login01 fortran_demo1]$ gfortran Solar_elevation_angle.f90 -o Solar_elevati
on_angle.x -L. -lsea
[ese-niecl@login01 fortran_demo1]$ ./Solar_elevation_angle.x
 delta =   -23.1656399
 h =   -28.4088993
 SEA =    36.6231003
```