

1.1

```
program Main

implicit none

integer :: i,u,z,x,y,j
real(4), dimension(:,:), allocatable :: m
real(4), dimension(:,:), allocatable :: n
real(4) :: r(5,5)

u=1
z=2
x=3
y=5

!read the M.dat
open(unit=u, file='M.dat', status='old')
allocate( m(y,x))

do i = 1,y
    read(u,*) (m(i,j),j=1,x)
enddo

close(u)

write(*,*) "N="
do i = 1,y
    write(*,'(5f8.2)') (m(i,:))
enddo

!read the N.dat
open(unit=z, file='N.dat', status='old')
allocate( n(x,y))

do i = 1,x
    read(z,*) (n(i,j),j=1,y)
enddo

close(z)

write(*,*) "M="
do i = 1,x
    write(*,'(5f8.2)') (n(i,:))
enddo

!call the subroutine
call Matrix_multip(m,n,r)
write(*,*) "M*N="
write(*,'(5f8.2)') r

deallocate(m,n)

u=50
open(unit=u, file='MN.dat', status='replace')
write(u,'(5f9.2)') r
close(u)

end program Main
```

1.2

```
! This is a subroutine
!-----
subroutine Matrix_multip(m,n,r)

implicit none

real(4), intent(in) ,dimension(:,:) :: m(5,3)
real(4), intent(in) ,dimension(:,:) :: n(3,5)
real(4), intent(out) ,dimension(:,:) :: r(5,5)

r=matmul(m,n)

end subroutine Matrix_multip
```

1.3

```
[ese-zhouyq@login01 fortran_demo1]$ gfortran Matrix_multip.f90 Main.f90 -o Main.x
[ese-zhouyq@login01 fortran_demo1]$ ./Main.x
```

```
N=
 19.48   15.79   19.28
 19.28   12.92   15.86
 15.86   11.29   14.04
 11.93   18.60   18.23
 19.28   12.92   15.86

M=
  7.72   4.11   1.44   4.80   5.55
  5.55   4.80   4.04   0.59   8.58
  0.59   8.58   2.26   7.72   4.11

M*N=
 249.40  229.90  193.38  206.09  229.90
 321.28  277.34  239.84  294.73  277.34
 135.42  115.80  100.18  133.52  115.80
 251.66  222.61  191.18  208.97  222.61
 322.83  283.04  242.60  300.72  283.04
```

```
[ese-zhouyq@login01 fortran_demo1]$ ll
total 520
-rwxr-xr-x 1 ese-zhouyq ese-ouycc 8816 Dec  8 18:59 a.out
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  125 Dec  8 18:36 DoLoopTest.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  301 Dec  8 18:36 DoWhileTest.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc   66 Dec  8 18:36 HelloWorld.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc 8816 Dec  8 19:05 HelloWorld.x
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  301 Dec  8 18:36 IfElseTest.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  230 Dec  8 18:36 ImplicitTypeTest.f90
-rw-r--r-- 1 ese-zhouyq ese-ouycc  766 Dec 22 15:27 Main.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc 17944 Dec 22 15:27 Main.x
-rw-r--r-- 1 ese-zhouyq ese-ouycc  280 Dec 22 15:15 Matrix_multip.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc   91 Dec  8 18:36 M.dat
-rw-r--r-- 1 ese-zhouyq ese-ouycc  230 Dec 22 15:27 MN.dat
-rwxr-xr-x 1 ese-zhouyq ese-ouycc   76 Dec  8 18:36 N.dat
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  410 Dec  8 18:36 PrecisionTest.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  183 Dec  8 18:36 TestArray.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  166 Dec  8 20:29 TestLeapYear.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc 8896 Dec  8 20:23 TestLeapYear.x
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  303 Dec  8 18:36 TestRelationalOps.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  182 Dec  8 18:36 TestUndeclared.f90
-rwxr-xr-x 1 ese-zhouyq ese-ouycc  449 Dec  8 18:36 VariableShowcase.f90
[ese-zhouyq@login01 fortran_demo1]$ vi MN.dat
```

MN.dat contains:

249.40	229.90	193.38	206.09	229.90
321.28	277.34	239.84	294.73	277.34
135.42	115.80	100.18	133.52	115.80
251.66	222.61	191.18	208.97	222.61
322.83	283.04	242.60	300.72	283.04

2.1

```

module Declination_angle

implicit none

    integer :: d
    real(8) :: a,b,pi

contains
    subroutine cal()
        pi=3.14159265
        write(*,*) 'Input the number of days since Jan. 1st d'
        read(*,*) d

        b=COS(pi/180*(360/365.24)*(d+10)+(360/pi)*0.0167*SIN((pi/180*360/365.24)*(d-2)))

        a=(ASIN(SIN(-23.44*pi/180)*b))*180/pi

    end subroutine cal
end module Declination_angle

```

2.2

```

module Solar_hour_angle

    real(4) :: h,LST
    contains
    subroutine cal2()
        write(*,*) 'Input the local solar time(in min) LST'
        read(*,*) LST

        h=15*((LST/60)-12)

    end subroutine cal2
end module Solar_hour_angle

```

2.3

```

program Solar_elevation_angle

use declination_angle
use solar_hour_angle

implicit none

real(4) :: SEA, L

write(*,*) 'Input latitude L'
read(*,*) L

call cal()
call cal2()

SEA=(ASIN(SIN(L*pi/180)*SIN(a*pi/180)+COS(L*pi/180)*COS(a*pi/180)*COS(h*pi/180)))*180/pi

print*, "Declination_angle = ", a
print*, "Solar_hour_angle = ", h
print*, "Solar_elevation_angle = ", SEA

end program Solar_elevation_angle

```

2,4

```

[ese-zhouyq@login01 fortran_demo1]$ gfortran -c Declination_angle.f90
[ese-zhouyq@login01 fortran_demo1]$ gfortran -c Solar_hour_angle.f90
[ese-zhouyq@login01 fortran_demo1]$ gfortran -c Solar_elevation_angle.f90
[ese-zhouyq@login01 fortran_demo1]$ ar rcvf libsea.a Declination_angle.o Solar_h
our_angle.o
a - Declination_angle.o
a - Solar_hour_angle.o
[ese-zhouyq@login01 fortran_demo1]$ gfortran Solar_elevation_angle.f90 -o Solar_
elevation_angle_lib.x -L. -lsea
[ese-zhouyq@login01 fortran_demo1]$ ./Solar_elevation_angle_lib.x
Input latitude L
22.542883
Input the number of days since Jan. 1st d
364
Input the local solar time(in min) LST
632
Declination_angle = -23.415861463273444
Solar_hour_angle = -21.9999924
Solar_elevation_angle = 39.3060265

```

Citation: Thanks to my classmates 邓伟豪, 伍日昕和谢栋 for teaching me codes.