Assignment 06

王帅　12132222

这次作业太难了，用了好几天，希望老师手下留情

1.1 [5 points] Write a program Main.f90 to read fortran_demo1/M.dat as the matrix M , and fortran_demo1/N.dat as the matrix N .

```
program READ

implicit none

integer                                                    :: u1, u2, a, b, c, d, i, j
real(4), dimension(:,:),allocatable    :: M, N


u1=111
u2=222
a=3
b=5
c=5
d=3


open(unit=u1,file='M.dat',status='old')
open(unit=u2,file='N.dat',status='old')


allocate(M(a,b))
allocate(N(a,b))

do i=1,b
    read(u1,*) M(i,:)
enddo


do i=1,d
    read(u2,*) N(i,:)
enddo


do i=1,b
    write(*,*) "Line ",i,":",M(i,:)
enddo



do i=1,d
    write(*,*) "Line ",i,":",N(i,:)
enddo


deallocate(M)
deallocate(N)


End program READ
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran Main.f90 -o Main.x
[ese-wangs@login02 fortran_demo1]$ ./Main.x
 Line           1 :   19.4799995       15.7900000       19.2800007
 Line           2 :   19.2800007       12.9200001       15.8599997
 Line           3 :   15.8599997       11.2900000       14.0400000
 Line           4 :   11.9300003       18.6000004       18.2299995
 Line           1 :   7.71999979       4.11000013       1.44000006       4.80000019
 Line           2 :   5.55000019       4.80000019       4.03999996      0.589999974
 Line           3 :  0.589999974       8.57999992       2.25999999       7.71999979
[ese-wangs@login02 fortran_demo1]$
```

1.2 [5 points] Write a subroutine Matrix_multip.f90 to do matrix multiplication.

```
subroutine Matrix_multiple(M,N,MN)

implicit none

real(8),dimension(4,3),intent(in) :: M
real(8),dimension(3,4),intent(in) :: N
real(8),dimension(4,4),intent(out) :: MN
integer :: i,j,k
real(8) :: a

do i=1,4
    do j=1,4
        a=0
        do k=1,3
            a=a+M(i,k)*N(k,j)
        enddo
        MN(i,j)=a
    enddo
enddo


end subroutine Matrix_multiple
```

1.3 [5 points] Call the subroutine Matrix_multip() from Main.f90 to compute M*N ; write the output to a new file MN.dat , values are in formats of f9.2 .

第三题需要将第二题中写好的 subroutine 带入到第一题中的 Main.f90 主程序中，并调用 subroutine 进行矩阵相乘的计算，得到结果。并且将计算得到的结果存入到新生成的 MN.dat 文件中。

```
program READ

implicit none

integer                                  :: u1, u2, a, b, c, d, i, j
real(4), dimension(:,:),allocatable      :: M, N
real(4),  dimension(4,4)  :: MN

u1=111
u2=222
a=3
b=4
c=4
d=3

open(unit=u1,file='M.dat',status='old')
open(unit=u2,file='N.dat',status='old')

allocate(M(b,a))
allocate(N(d,c))

do i=1,b
    read(u1,*) M(i,:)
enddo

do i=1,d
    read(u2,*) N(i,:)
enddo

close(u1)
close(u2)

do i=1,b
    write(*,*) "Line ",i,":",M(i,:)
enddo

do i=1,d
    write(*,*) "Line ",i,":",N(i,:)
enddo

call  Matrix_multiple(M,N,MN)
```

```
do i=1,4
    write(*,*) "Line ",i,":",MN(i,:)
enddo

open(unit=u1,file='MN.dat',status='replace')
do i=1,4
    write(u1,'(f9.2)') MN(i,:)
enddo

close(u1)

deallocate(M)
deallocate(N)

End program READ


!this is the subroutine

subroutine Matrix_multiple(M,N,MN)

implicit none

real(4),dimension(4,3),intent(in) :: M
real(4),dimension(3,4),intent(in) :: N
real(4),dimension(4,4),intent(out) :: MN
integer :: i,j,k
real(4) :: a

do i=1,4
    do j=1,4
        a=0
        do k=1,3
            a=a+M(i,k)*N(k,j)
        enddo
        MN(i,j)=a
    enddo
enddo

end subroutine Matrix_multiple
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran Multiple.f90 -o Multiple.x
[ese-wangs@login02 fortran_demo1]$ ./Multiple.x
 Line           1 :   19.4799995       15.7900000       19.2800007
 Line           2 :   19.2800007       12.9200001       15.8599997
 Line           3 :   15.8599997       11.2900000       14.0400000
 Line           4 :   11.9300003       18.6000004       18.2299995
 Line           1 :   7.71999979       4.11000013       1.44000006       4.80000019
 Line           2 :   5.55000019       4.80000019       4.03999996      0.589999974
 Line           3 :  0.589999974       8.57999992       2.25999999       7.71999979
 Line           1 :   249.395294       321.277222       135.415604       251.661697
 Line           2 :   229.904999       277.335602       115.803604       222.606003
 Line           3 :   193.382294       239.839798       100.180397       191.177887
 Line           4 :   206.085297       294.725708       133.522995       208.973602
[ese-wangs@login02 fortran_demo1]$
```

```
249.40
321.28
135.42
251.66
229.90
277.34
115.80
222.61
193.38
239.84
100.18
191.18
206.09
294.73
133.52
208.97
```

2.1 [5 points] Write a module `Declination_angle` that calculates the *declination angle* on a given date.

这个题我先写了一个 module，然后仿照老师上课的 test 写了一个用来调用 module 的文件进行计算，最后选择了 11.08 作为检验的时间，得到的结果是-15.88276。

```
module Declination_angle

implicit none

real, parameter :: pi=3.1415926

contains

    subroutine calculate_angle(mon,day,angle)

    integer,intent(in) :: mon, day
    real(8),intent(out) :: angle
    integer ::a
    a=(mon-1)*30+day

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

    end subroutine calculate_angle

end module Declination_angle
```

```
program TestProgram

use Declination_angle

implicit none

real(8) ::angle
integer ::mon, day

mon=11
day=08

call calculate_angle(mon,day,angle)

write(*,*) angle

end program TestProgram
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran test_Declination_angle.f90 Declination_angle.f90 -o test.x
[ese-wangs@login02 fortran_demo1]$ ./test.x
  -15.882760442335213
[ese-wangs@login02 fortran_demo1]$
```

2.2 [10 points] Write a module Solar_hour_angle that calculates the solar hour angle in a given location for a given date and time.

这个题个上一个题目思路类似，就是计算公式特别麻烦。检验的用的是 11.08 深圳下午两点作为标准，计算出来结果。

```
module solar_hour_angle

implicit none

real, parameter :: pi=3.1415926

contains

    subroutine calculation(lon,mon,day,t,angle)
    implicit none

    integer,intent(in) :: mon, day
    real(4),intent(in) :: lon, t
    real(4),intent(out) :: angle
    integer :: a
    real(4) :: offset, eot, gama

    a=(mon-1)*30+day
    gama=2*pi/365*(a-1+(t-12)/24)
    eot=229.18*(0.000075+0.001868*cos(gama)-0.032077*sin(gama)-0.014615*cos(2*gama)-0.040849*sin(2*gama))
    offset=eot+MOD(lon,15.0)
    angle=15*(t-12)+offset/60

    end subroutine calculation

end module solar_hour_angle
```

```
program Test

use solar_hour_angle

implicit none

real(4) :: t,lon,angle
integer :: mon,day

t=14
lon=114.05
mon=11
day=08

call calculation(lon,mon,day,t,angle)

write(*,*) angle

end program Test
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran test_solar_hour_angle.f90 solar_hour_angle.f90 -o test2.x
[ese-wangs@login02 fortran_demo1]$ ./test2.x
   30.4230843
```

2.3 [5 points] Write a main program (Solar_elevation_angle.f90) that uses module Declination_angle and Solar_hour_angle to calculate and print the SEA in a given location for a given date and time.

第三题需要调用前两题的 module，然后调用的时候需要保证变量的一致性，以及不同变量如 pi，多次调用的时候名称不能一致的问题，需要注意。最后检验选择了北纬 24.24，东经 24.24 在 11.08 日上午八点的结果。

```
program SEA

use Declination_angle
use solar_hour_angle

implicit none

real, parameter :: newpi=3.1415926
real(4) :: lat,lon,t,angle,dangle
integer :: mon,day
real(4) :: a

lat=24.24
lon=24.24
t=8
mon=11
day=08

call calculate_angle(mon,day,dangle)


call calculation(lon,mon,day,t,angle)


a=asin(sin(lat/180*newpi)*sin(dangle/180*newpi)+cos(lat/180*newpi)*cos(dangle/180*newpi)*cos(angle/180*newpi))
a=a/newpi*180.0
write(*,*) a

end program SEA
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran Solar_elevation_angle.f90 Declination_angle.f90 solar_hour_angle.f90 -o te
st3.x
[ese-wangs@login02 fortran_demo1]$ ./test3.x
   19.3777027
```

2.4 [5 points] Create a library (libsea.a) that contains Declination_angle.o and Solar_hour_angle.o . Compile Solar_elevation_angle.f90 using libsolar.a . Print the SEA for Shenzhen (22.542883N, 114.062996E) at 10:32 (Beijing time; UTC+8) on 2021-12-31 .

```
program SEA

use Declination_angle
use solar_hour_angle

implicit none

real, parameter :: newpi=3.1415926
real(4) :: lat,lon,t,angle,dangle
integer :: mon,day
real(4) :: a

lat=22.542883
lon=114.062996
t=10.0+32/60
mon=12
day=31

call calculate_angle(mon,day,dangle)


call calculation(lon,mon,day,t,angle)


a=asin(sin(lat/180*newpi)*sin(dangle/180*newpi)+cos(lat/180*newpi)*cos(dangle/180*newpi)*cos(angle/180*newpi))
a=a/newpi*180.0
write(*,*) a

end program SEA
```

```
[ese-wangs@login02 fortran_demo1]$ gfortran -c Declination_angle.f90
[ese-wangs@login02 fortran_demo1]$ gfortran -c solar_hour_angle.f90
[ese-wangs@login02 fortran_demo1]$ ar rcvf libsea.a Declination_angle.o solar_hour_angle.o
a - Declination_angle.o
a - solar_hour_angle.o
[ese-wangs@login02 fortran_demo1]$ gfortran SHENZHEN.f90 -o test4.x -L -lsea
```

```
[ese-wangs@login02 fortran_demo1]$ ./test3.x
   35.7903099
```