```
!Preprocessing gravity data
!Programmed by Li Haosi
!Class: 2016260202
!Completion data: 2019.04.11
!-------------------------------------------------------------------------------!
!num_x:  the number of  x nodes        num_y: the number of y nodes
!y_base: the y of the basic point          height_base: the height of  the basic point
ave_latitude: the average latitude of  this whole area
!density: the average density of the medium layer          x_max: the maximum for x
y_max: the maximum for y
!height(num_x, num_y): every node's height          landscapefile: the file recording the
landscape        cmdfile: the file storing the parametres
!-------------------------------------------------------------------------------!
program main

implicit none
integer num_x, num_y, i, j
real y_base, height_base, ave_latitude, density
real x_max, x_min, y_max, y_min
real, allocatable:: height(:, :)
character*80 landscape_file, cmdfile, g_phi_file, g_m_file, g_h_file

call input_par(cmdfile, y_base, height_base, ave_latitude, density, g_phi_file, g_m_file, g_h_file,
landscape_file)
!-------------------------------------------------------------------------------!
open(30, file = landscape_file)
read(30, *)
read(30, *) num_x, num_y
read(30, *) x_min, x_max
read(30, *) y_min, y_max
read(30, *)
allocate(height(num_x, num_y))
read(30, *) ((height(i, j), j = 1, num_y), i = 1, num_x)
close(30)
!-------------------------------------------------------------------------------!
call  latitude_adjust(ave_latitude,  height_base,  height,  x_min,  x_max,  y_max,  y_min,  num_x,
num_y, y_base, g_phi_file)
call mlayer_adjust(density, height, height_base, num_x, num_y, g_m_file, x_min, x_max, y_max,
y_min)
call height_adjust(height, height_base, num_x, num_y, g_h_file, x_min, x_max, y_max, y_min)
deallocate(height)
end program main

!read and get the parametres in cmd.par
```

```
subroutine input_par(cmdfile, y_base, height_base, ave_latitude, density, g_phi_file, g_m_file, g_h_file, landscape_file)
    real y_base, height_base, ave_latitude, density
    character*(80) cmdfile,landscape_file, g_phi_file, g_m_file, g_h_file, temp

    cmdfile = 'cmd.par'
    open(20, file = cmdfile)
    read(20, *) landscape_file, temp
    read(20, *) height_base, temp
    read(20, *) y_base, temp
    read(20, *) ave_latitude, temp
    read(20, *) density, temp
    read(20, *) g_phi_file, temp
    read(20, *) g_m_file, temp
    read(20, *) g_h_file, temp
    close(20)
end subroutine input_par

!Calculate the adjusted value for latitude
subroutine latitude_adjust(ave_latitude, height_base, height, x_min, x_max, y_max, y_min, num_x, num_y, y_base, g_phi_file)
    integer num_x, num_y, i, j, k
    character*(20) g_phi_file
    real:: ave_latitue, x_max, x_min, y_max, y_min, interval_x, interval_y, y_base
    real:: height(num_x, num_y), distance(num_x, num_y), g_phi(num_x, num_y)
    interval_x = (x_max - x_min)/(num_x-1)
    interval_y = (y_max - y_min)/(num_y-1)
    do i = 1, num_y
        do j = 1, num_x
            if (i > y_base) then
                distance(i, j) = -(y_base/interval_y + 1 - i)*interval_y
            else
                distance(i, j) = (y_base/interval_y + 1 - i)*interval_y
            end if
            write(*, *) distance(i, j), y_base/interval_y
            g_phi(i, j) = -8.14*sin(2*ave_latitude*3.14159256/180)*distance(i, j)
        end do
    end do
    max_phi = maxval(g_phi)
    min_phi = minval(g_phi)
    !Output the data
    open(41, file = g_phi_file)
    write(41, "(a)") 'DSAA'
    write(41, *)num_x, num_y
```

```
        write(41, *)x_min, x_max
        write(41, *)y_min, y_max
        write(41, *)min_phi, max_phi
        do i = 1, num_x
            write(41, *) (g_phi(i, j), j = 1, num_y)
        end do
        close(41)
end subroutine

!Calculate the adjusted value for medium layer
subroutine mlayer_adjust(density, height, height_base, num_x, num_y, g_m_file, x_min, x_max,
y_max, y_min)
        integer num_x, num_y, i, j
        real height_base, height(num_x, num_y), g_m(num_x, num_y), x_min, x_max, y_max,
y_min
        character*(20) g_m_file
        interval_x = (x_max - x_min)/(num_x-1)
        interval_y = (y_max - y_min)/(num_y-1)
        do i = 1, num_x
            do j = i, num_y
                g_m(i, j) = -(0.419 - 0.2095/167000*(height(i, j) - height_base)) * density *
(height(i, j) - height_base)
            end do
        end do
        max_m = maxval(g_m)
        min_m = minval(g_m)
        !Output the data
        open(50, file = g_m_file)
        write(50, "(a)") 'DSAA'
        write(50, *)num_x, num_y
        write(50, *)x_min, x_max
        write(50, *)y_min, y_max
        write(50, *)min_m, max_m
        do i = 1, num_x
            write(50, *) (g_m(i, j), j = 1, num_y)
        end do
        close(50)
end subroutine

!Calculate the adjusted value for height
subroutine height_adjust(height, height_base, num_x, num_y, g_h_file, x_min, x_max, y_max,
y_min)
        integer num_x, num_y, i, j
        real g_h(num_x, num_y), height(num_x, num_y)
```

```
    real height_base, x_min, x_max, y_max, y_min
    character *(20) g_h_file
    interval_x = (x_max - x_min)/(num_x-1)
    interval_y = (y_max - y_min)/(num_y-1)
    do i = 1, num_x
        do j = 1, num_y
            g_h(i, j) = 3.086 * (height(i, j) - height_base)
        end do
    end do
    max_h = maxval(g_h)
    min_h = minval(g_h)
    !Output the data
    open(60, file = g_h_file)
    write(60, "(a)") 'DSAA'
    write(60, *)num_x, num_y
    write(60, *)x_min, x_max
    write(60, *)y_min, y_max
    write(60, *)min_h, max_h
    do i = 1, num_x
        write(60, *) (g_h(i, j), j = 1, num_y)
    end do
    close(60)
end subroutine
```