PROGRAM prinsim

IMPLICIT NONE

! Declaración de variables y arreglos

REAL, ALLOCATABLE :: T( : ), X( : ), Y( : )

! T = arreglo para tiempo

! X = arreglo para valores de x

! Y = arreglo para valores de y

REAL :: vo, voy, vox, xo, yo, tt, O, h

! vo = velocidad incial

! voy = velocidad inicial en y

! vox = velocidad inicial en x

! xo = posición inicial en x

! yo = posición inicial en y

! tt = tiempo total

! O = ángulo

INTEGER :: n, i

! número de datos (particiones)

CHARACTER(len=1) :: ang

PRINT\*, "Introduzca la posicion inicial (xo)"

READ\*, xo

PRINT\*, "Introduzca la posicion inicial (yo)"

READ\*, yo

PRINT\*, "Ingrese la velocidad inicial."

READ\*, vo

PRINT\*, "Ingrese el angulo de proyección."

print\*, "¿Prefiere en grados (G) o en radianes(R)?"

READ\*, ang

! Select Case que le presenta al usuario las opciones disponibles

angulo: SELECT CASE (ang)

CASE("g","G")

! El usuario prefiere en grados

PRINT\*, "Introduce el angulo"

READ\*, O

O=(O\*3.14159)/180

CASE("r","R")

! El usuario prefiere en radianes

PRINT\*, "Introduce el angulo"

READ\*, O

END SELECT angulo

! Ecuaciones para obtener las velocidades iniciales en x & y

vox = vo\*(COS(O))

voy = vo\*(SIN(O))

CALL subsim(yo, voy, tt)

! dummy variables: yo, voy, tt

PRINT\*, "¿Cuál es el número de datos (ancho de paso) que desea usar?"

READ\*, n

h = tt/n

ALLOCATE(X(1:n), Y(1:n), T(1:n))

! creando los arreglos de X, Y, T que almacenaran los valores correspondientes

! LOOP que nos ayuda a obtener los elementos de cada arreglo

DO i=1,n

T(i)= i\*h

X(i) = xo + vox\*T(i)

Y(i) = yo + voy\*T(i) -((9.81)/2)\*T(i)\*\*2

END DO

CALL freefall(n,X,Y,T)

! Se le llama al script como si fuera una subrutina

END PROGRAM prinsim

SUBROUTINE subsim(yo, voy, tt)

! La subrutina se encarga de obtener el valor del tiempo total usando la ecuación general, el tiempo mayor sera el valor del tiempo total

IMPLICIT NONE

REAL, INTENT(IN) :: yo, voy

REAL, INTENT(INOUT) :: tt

REAL :: t1, t2

! Ecuación General

t2 = (-voy - sqrt(voy\*\*2 -4\*(-9.81/2)\*(yo)))/(-9.81)

t1 = (-voy + sqrt(voy\*\*2 -4\*(-9.81/2)\*(yo)))/(-9.81)

PRINT\*, t1, t2

! IF que nos ayuda a determinar cual va a ser el valor del tiempo total, dependiendo de cual tiempo sea mayor

IF(t1>t2)THEN

tt=t1

ELSE

tt=t2

END IF

END SUBROUTINE subsim