

1.

```

program stima
  implicit none
  real,dimension(:),allocatable :: x,y,sx,sy,r,sr
  real :: s00,s01,s11,s10,s20,m,q,t,rm,a,b
  integer :: n,i,error

  print*,"Inserisci il numero totale dei valori"      ! Numero di valori
  read*,n

  allocate(x(n),y(n),sx(n),sy(n),r(n),sr(n),stat = error)  ! Allocations array
  if(error /= 0)then
    print*, " Non c'e' abbastanza spazio su disco"
    print*, " Il programma verra' terminato"
    stop
  end if

  do i = 1,n,1                                           ! Inserimento dei valori
    print*, " Riga n*",i
    print*, " Inserisci il valore di x"
    read*,x(i)
    print*, " Inserisci il valore di y"
    read*,y(i)
    print*, " Inserisci il valore di sigma x"
    read*,sx(i)
    print*, " Inserisci il valore di sigma y"
    read*,sy(i)
    print*
  end do

  s00 = 0.0                                              ! Calcola i vari s
  s01 = 0.0
  s11 = 0.0
  s10 = 0.0
  s20 = 0.0

  do i = 1,n,1
    s00 = s00 + (1.0 / sy(i)**2)
    s01 = s01 + (y(i) / sy(i)**2)
    s11 = s11 + (x(i)*y(i) / sy(i)**2)
    s10 = s10 + (x(i) / sy(i)**2)
    s20 = s20 + (x(i)**2 / sy(i)**2)
  end do

  m = (s00*s11 - s10*s01) / (s00*s20 - s10**2)         ! Calcola m e q
  q = (s20*s01 - s10*s11) / (s00*s20 - s10**2)

  a = 0.0
  b = 0.0

  do i = 1,n,1                                           ! Calcola i vari r(i) e relativo sigma quadro
    r(i) = x(i) / y(i)
    sr(i) = (1.0 / y(i)**2)*sx(i)**2 + (x(i)**2 / y(i)**2)*sy(i)**2
    a = a + (r(i) / sr(i))
    b = b + (1.0 / sr(i))
  end do

  rm = a / b

```

```

print*
print*, " Paragone tra la m e media ponderata di resistenze"
print*, " Valore di 1/m: ", 1.0/m
print*, " Valore di r : ", rm
print*

```

```

do i = 1,n,1
    t = m*x(i) + q
    write(1,*)x(i),t
    write(2,*)x(i),y(i),sx(i),sy(i)
end do

```

! Scrive su file, calcola la retta

end program stima

Valore di $1/m$: 58.885380 [calcolato]

Valore di R: 45.977829 [da media ponderata]

