

## Laboratorio 2 - esercitazione 1

*soluzioni di Italo Ossena*

1.

program grafico

implicit none

real,dimension(6) :: x,y,ex,ey

integer :: i

x(1) = 1.0                   ! Valori

x(2) = 3.0

x(3) = 5.0

x(4) = 10.0

x(5) = 15.0

x(6) = 20.0

ex(1) = 0.010

ex(2) = 0.010

ex(3) = 0.020

ex(4) = 0.020

ex(5) = 0.050

ex(6) = 0.050

y(1) = 7.50

y(2) = 5.30

y(3) = 4.10

y(4) = 1.460

y(5) = 0.380

y(6) = 0.1890

ey(1) = 0.80

ey(2) = 0.50

ey(3) = 0.40

ey(4) = 0.130

ey(5) = 0.050

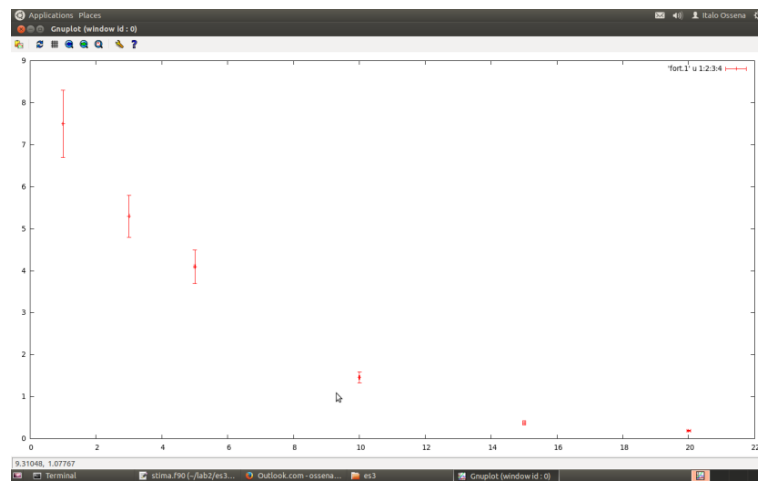
ey(6) = 0.0180

do i = 1,6,1

  write(unit=1,fmt=\*)x(i),y(i),ex(i),ey(i)

end do

end program grafico



2. a

```
program istogramma
  implicit none
  integer :: s,t,error,i,np,nm,n_int,w
  integer, dimension(:),allocatable :: seme
  integer, dimension(50) :: n
  real, dimension(200) :: x
  real, dimension(10000) :: y
  real, parameter :: h = 0.02
  real :: l
  real :: j,k,ex,sig,ermin,ermax

  ! Uso di un seed per generare scelte casuali
  print*, " Il programma usa un seed per rendere le scelte del computer casuali ogni volta"
  call random_seed(size=i)
  print*, " Bisogna inserire",i,"numeri del seed"
  print*, " "
  allocate(seme(i),stat=error)
  if(error /= 0)then
    print*, " "
    print*, " Non c'e' spazio nella memoria per mettere tutti questi valori"
    print*, " Il programma verra' terminato"
    stop
  end if
  read*,seme
  call random_seed(put=seme)

  nm = 200
  np = 10000
  n_int = 50

  do i = 1,nm,1
    call random_number(j)
    x(i) = j
  end do

  do i = 1,np,1
    call random_number(k)
    y(i) = k
  end do

  n = 0

  do s = 1,nm,1
    do t = 1,n_int,1
      l = h * t
      if (x(s) <= l)then
        n(t) = n(t) + 1
        exit
      else
        cycle
      end if
    end do
  end do

  do t = 1,n_int,1
    l = h * t
    ex = (nm*1.0) * (1.0 / (n_int*1.0))
    sig = sqrt(ex * (1.0 - (1.0/(n_int*1.0))))
    ermin = ex - sig
    ermax = ex + sig
```

```

write(unit=1,fmt=*)l-h,n(t),ermin,ermax
end do

```

```

l = 1.0
w = 0
ermin = 0.0
ermax = 0.0
write(unit=1,fmt=*)l,w,ermin,ermax

```

```

n = 0

```

```

do s = 1,np,1
do t = 1,n_int,1
l = h * t
if (y(s) <= l)then
n(t) = n(t) + 1
exit
else
cycle
end if
end do
end do

```

```

do t = 1,n_int,1
! Creazione degli errori e scrittura su file
l = h * t
ex = (np*1.0) * (1.0 / (n_int*1.0))
sig = sqrt(ex * (1.0 - (1.0/( n_int*1.0))))
ermin = ex - sig
ermax = ex + sig
write(unit=2,fmt=*)l-h,n(t),ermin,ermax
end do

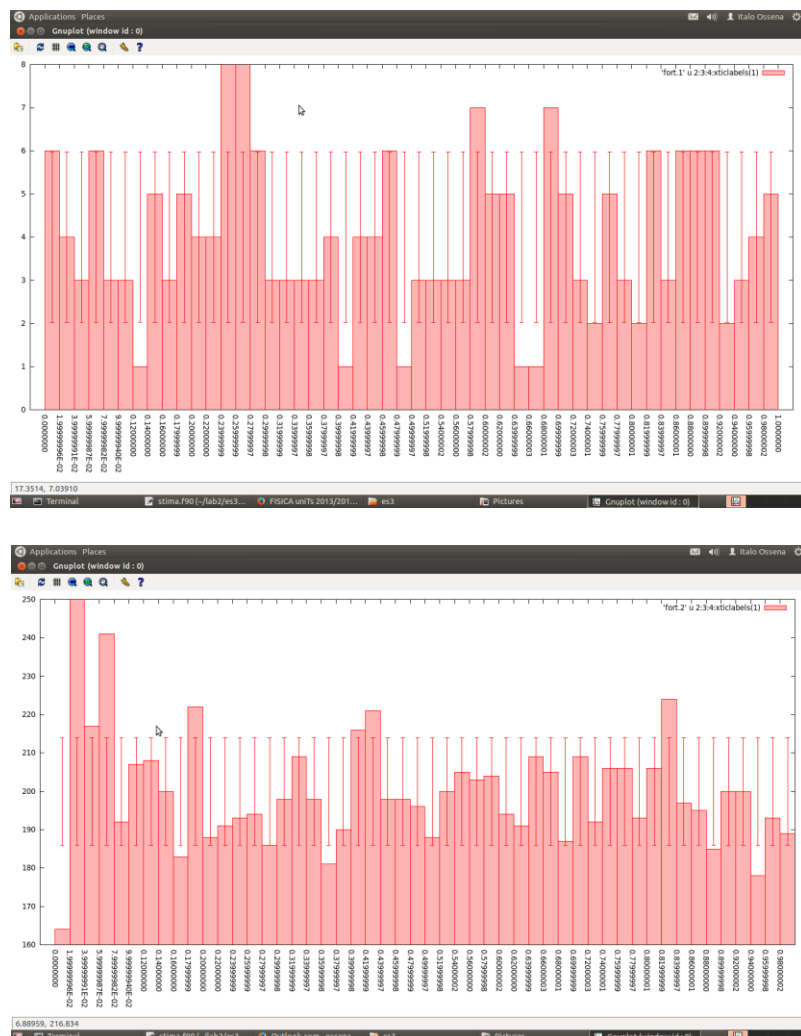
```

```

end program istogramma

```

seed usato: 1 2 3 4 5 6 7 8 9 1 2 3



## 2. b

```
program istogramma2
  implicit none
  integer :: s,t,error,i,np,nm,n_int
  integer, dimension(:),allocatable :: seme
  integer, dimension(50) :: n
  real, dimension(:),allocatable :: x
  real, dimension(:),allocatable :: y
  real :: l,h,a,b,p
  real :: j,k,ex,sig,ermin,ermax

  print*, " Il programma usa un seed per rendere le scelte del computer casuali ogni volta"
  call random_seed(size=i)
  print*, " Bisogna inserire",i,"numeri del seed"
  print*, " "
  allocate(seme(i),stat=error)
  if(error /= 0)then
    print*, " "
    print*, " Non c'e' spazio nella memoria per mettere tutti questi valori"
    print*, " Il programma verra' terminato"
    stop
  end if
  read*,seme
  call random_seed(put=seme)

  print*
  print*, " Inserisci gli estremi dell'intervallo"
  print*
  read*,a,b

  nm = 200
  np = 10000
  n_int = 50
  h = (b-a) / n_int*1.0
  allocate(x(nm),stat=error)      ! Allocazione dell'array
  if(error /= 0)then
    print*, " "
    print*, " Non c'e' spazio nella memoria per mettere tutti questi valori"
    print*, " Il programma verra' terminato"
    stop
  end if

  do i = 1,nm,1                  ! Generazione dei numeri secondo la formula (200 valori)
    call random_number(j)
    x(i) = 2.0*sqrt(j)
  end do

  n = 0

  do s = 1,nm,1                  ! Generazione colonne degli istogrammi
    do t = 1,n_int,1
      l = h * t
      if (x(s) <= l)then
        n(t) = n(t) + 1
        exit
      else
        cycle
      end if
    end do
  end do
```

```

    end do
end do

do t = 1,n_int,1          ! Calcolo degli errori e scrittura su foglio
    l = h * t
    p = ((l**2) - ((l-h)**2)) / 4.0
    ex = nm*p
    sig = sqrt(ex * (1.0 - p) / 8.0)
    ermin = ex - sig
    ermax = ex + sig
    write(unit=1,fmt=*)l,n(t),ermin,ermax
end do

deallocate(x)

allocate(y(np),stat=error)      ! Allocazione dell'array
if(error /= 0)then
    print*, " "
    print*, " Non c'e' spazio nella memoria per mettere tutti questi valori"
    print*, " Il programma verra' terminato"
    stop
end if

do i = 1,np,1                  ! Generazione dei valori secondo la formula (10000 valori)
    call random_number(k)
    y(i) = 2.0*sqrt(k)
end do

n = 0

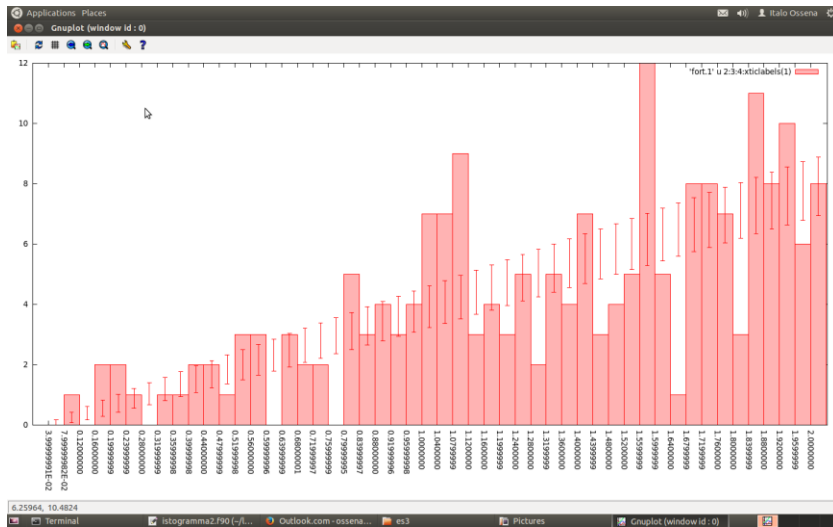
do s = 1,np,1                  ! Generazione colonne degli istogrammi
    do t = 1,n_int,1
        l = h * t
        if (y(s) <= l)then
            n(t) = n(t) + 1
            exit
        else
            cycle
        end if
    end do
end do

do t = 1,n_int,1              ! Calcolo degli errori e scrittura su foglio
    l = h * t
    p = ((l**2) - ((l-h)**2)) / 4.0
    ex = np*p
    sig = sqrt(ex * (1.0 - p) / 4.0)
    ermin = ex - sig
    ermax = ex + sig
    write(unit=2,fmt=*)l,n(t),ermin,ermax
end do

deallocate(y)

end program istogramma2

```



*seed usato: 1 2 3 4 5 6 7 8 9 1 2 3*

*estremi: 0.0 - 2.0*

