Laboratorio 2 - esercitazione 2 *soluzione di Italo Ossena*

1.

```
module funzioni
  contains

recursive function bin(n,k) result(binomio) ! Binomio di Newton
  implicit none
  integer,intent(in) :: n,k
  integer :: binomio

  if(k == 0)then
    binomio = 1
  else
    binomio = bin(n,k-1) * (n-k+1)/k
  end if
end function bin

recursive function fatt(s) result(fat)      ! Fattoriale
  implicit none
  integer,intent(in) :: s
  integer :: fat

  if(s <= 1)then
    fat = 1
  else
    fat = fatt(s - 1) * s
  end if
end function fatt

end module funzioni

program distribuzioni
  use funzioni
  implicit none
  integer :: n,k,nu,mu,i
  real :: p,q,pbin,ppos,pgau,x,t,pics,z,y,var
  real,parameter :: pi = acos(-1.0)

  print*," Binaria"                    ! Binomiale
  print*," Inserisci n"
  read*, n
  print*," Inserisci p"
  read*,p

  q = 1.0 - p

  pbin = 0.0

  do k = 0,n,1
    pbin = bin(n,k)*(p**k)*(q**(n-k))
    write(unit=1,fmt=*)k*1.0,pbin
  end do



  print*," Poissoniana"                  ! Poissoniana
  print*," Inserisci nu"
  read*,nu
```

```
ppos = 0.0

do k = 1,20,1
  ppos = (nu**k)*exp(-nu*1.0)/fatt(k)
  write(unit=2,fmt=*)k*1.0,ppos
end do



print*," gaussiana"                    ! Gaussiana
print*," Inserisci mu"
read*,mu
print*," Inserisci sigma quadro"
read*,var

pgau = 0.0

t = 0.1

do i = 0,2000,1
  x = -100.0 + (t*i)
  pgau = exp(-((x*1.0-mu)**2)/(2*var))/(sqrt(2*pi*var))
  write(unit=3,fmt=*)x,pgau
end do



print*," X**2"                         ! Chi quadro
print*," Inserisci n"
read*,n

pics = 0
y = (n*1.0)/2.0

do i = 0,2000,1
  z = 0.0 + (t*i)
  pics = (z**(y-1.0))*exp(-z/2.0)/((2**y)*gamma(y))
  write(unit=4,fmt=*)z,pics
end do

end program distribuzioni
```

valori usati:
- binomiale: n = 20 p = 0.1
- Poisson: nu = 2
- Gauss: mu = 2, sigma quadro = 2
- chi quadro: n = 2