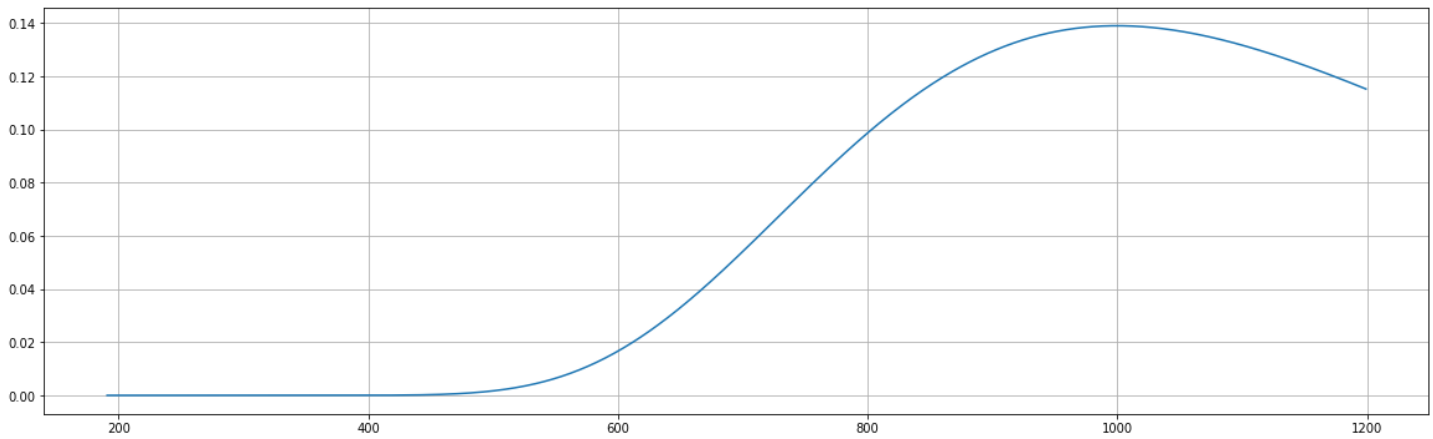
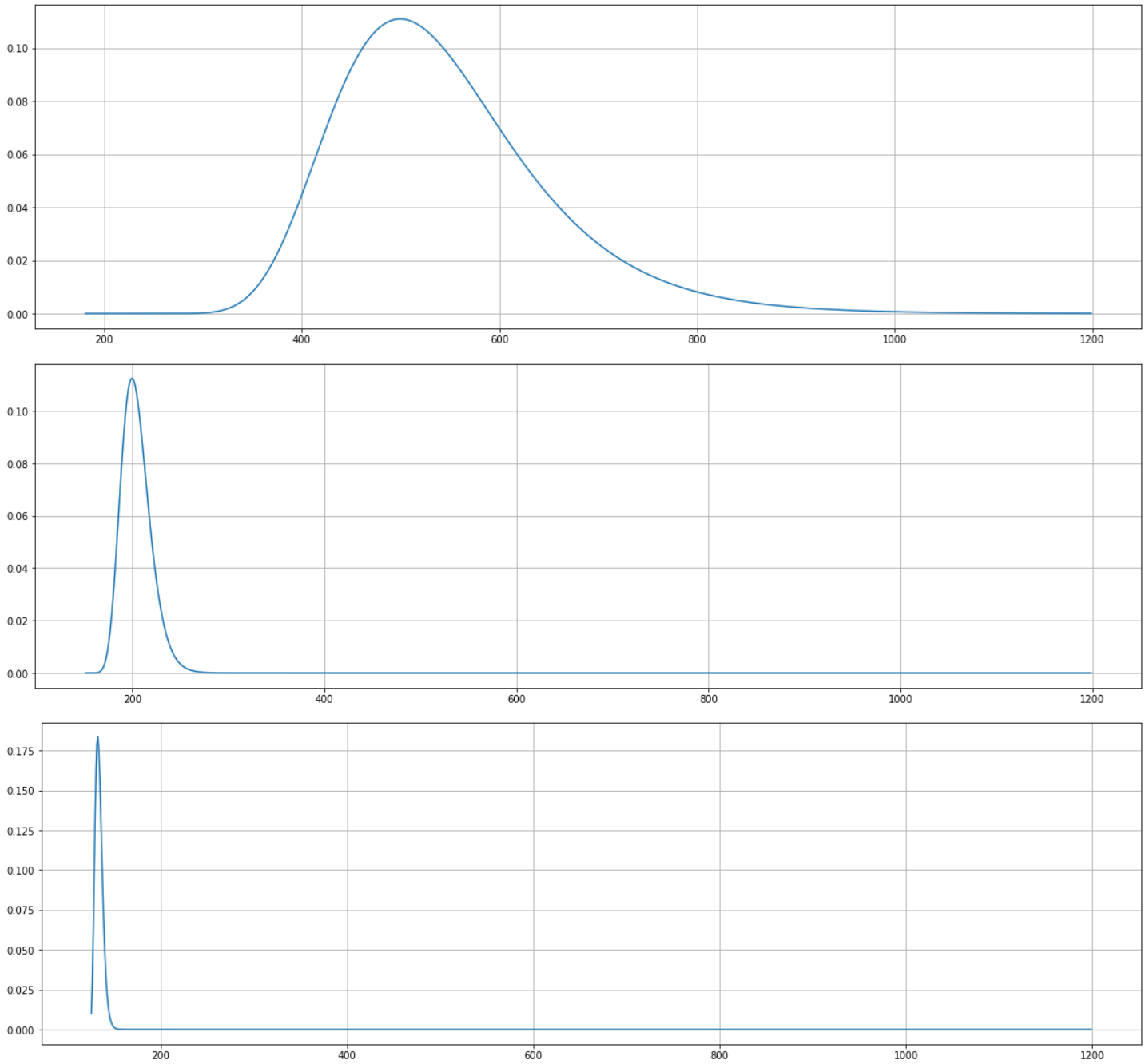
EE325 Assignment 2

By- Samar Agarwal, Vasu Soni, Tirth Joshi

In [3]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
m = 100
def maxprob(p):
  fm = math.factorial(m)
  fp = math.factorial(p)
  A = []

  for n in range (2*m-p+1,1200):
    b = n - 2*m + p
    a = n - m + p
    c = n - m
    d = m - p
    fa = math.factorial(a)
    fb = math.factorial(b)
    fc = math.factorial(c)
    fd = math.factorial(d)
    fn = math.factorial(n)
    fp = math.factorial(p)
    y = fm*fm*fc*fc/(fn*fp*fd*fd*fb)
    A.append(y)

  for i in range (1,len(A)-1):
    if(A[i]>A[i+1]):
      print(i + 2*m-p+1)
      break


  plt.figure(figsize=(20,6))
  B  = np.arange(0,len(A))
  plt.plot(np.arange(2*m-p+1,1200),A)
  plt.grid()

maxprob(10)
maxprob(20)
maxprob(50)
maxprob(75)
```

```
1000
500
200
133
```





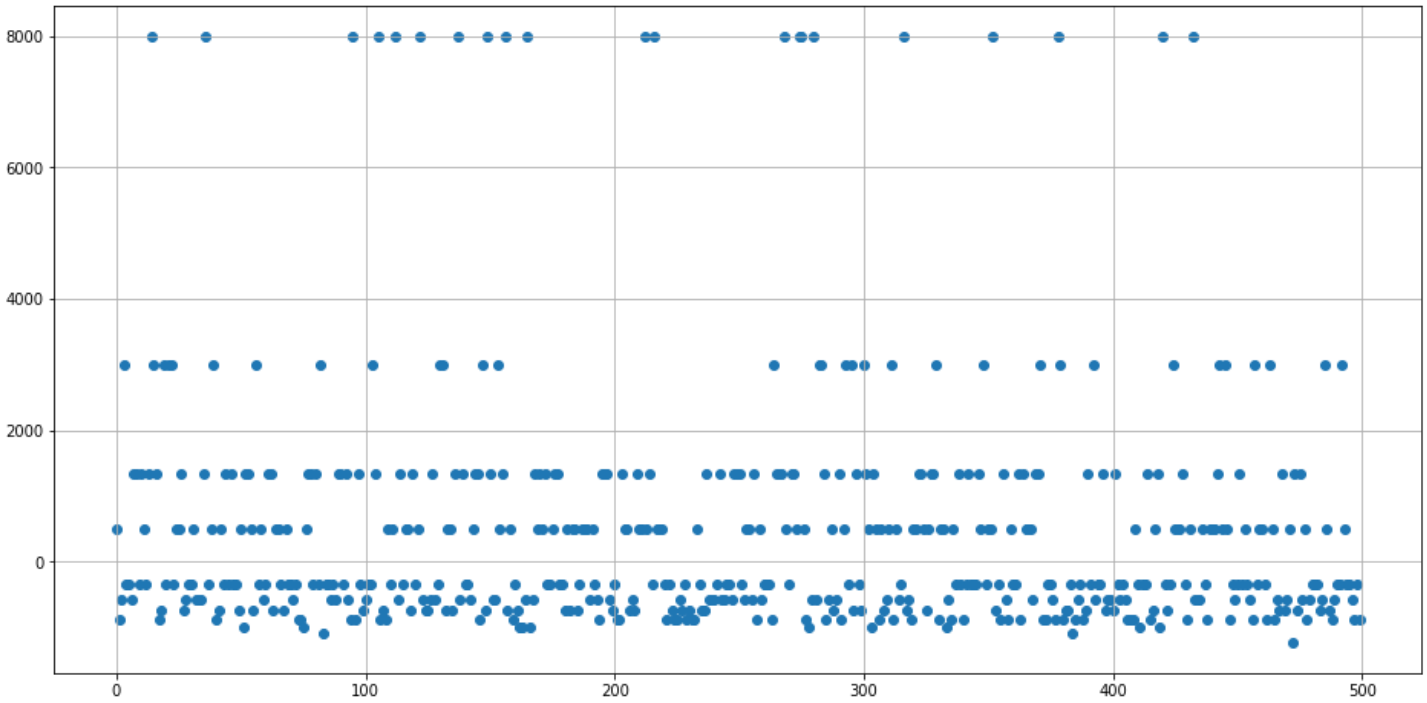
The given graphs are in the order p = 10,20,50,75.

Looking at the graphs, for each p we have obtained the local maxima for prob(p | n) which gives the value of n which has highest probability of being total no of fishes, for a given.As this n is most probable , for the given p we will guess this n to be total no of fishes.

In [4]:
```
import math
import numpy as np
import random
import matplotlib.pyplot as plt

def nCr(n,r):
  ans=1
  for i in range (r):
    ans*=n/(i+1)
    n-=1
  return ans
```

```
bestguesses = [10000,4999,3332,2499,1665,1427,1249,1110,1111]
def opbestguess(p):
  if p == 0:
    return bestguesses[0]
  if p<10 and p>0:
    return bestguesses[p-1]
  else :
    return bestguess(p)


def bestguess(p):

  m = 100
  fm = math.factorial(m)
  fp = math.factorial(p)
  A = []

  for n in range (200-p,3000):
    y = nCr(m,p)*nCr(n-m,m-p)/nCr(n,m)
    A.append(y)

  for i in range (1,len(A)-1):#2-4999 , 1 -10000
    if(A[i]>A[i+1]):
      return(i + 200-p)
      break

S = []
E = np.zeros(500)
for r in range(0,500):
  num = 0
  N = 2000
  pond = np.zeros(2000)
  y = random.sample(range(0,2000),100)
  for i in range(0,100):
    pond[y[i]] = 1
  y = random.sample(range(0,2000),100)
  count=0
  for i in y:
    if(pond[i]==1):
      count+=1
  My_N = opbestguess(count)
  S.append(My_N)
  error = (My_N - N)
  E[r]=error

print("The mean is " ,E.mean(),"\n\n")
print("The varience is ",np.var(E),"\n\n")
plt.figure(figsize=(16,8))
plt.grid()
plt.scatter(np.arange(0,E.shape[0]),E)
plt.show()
```

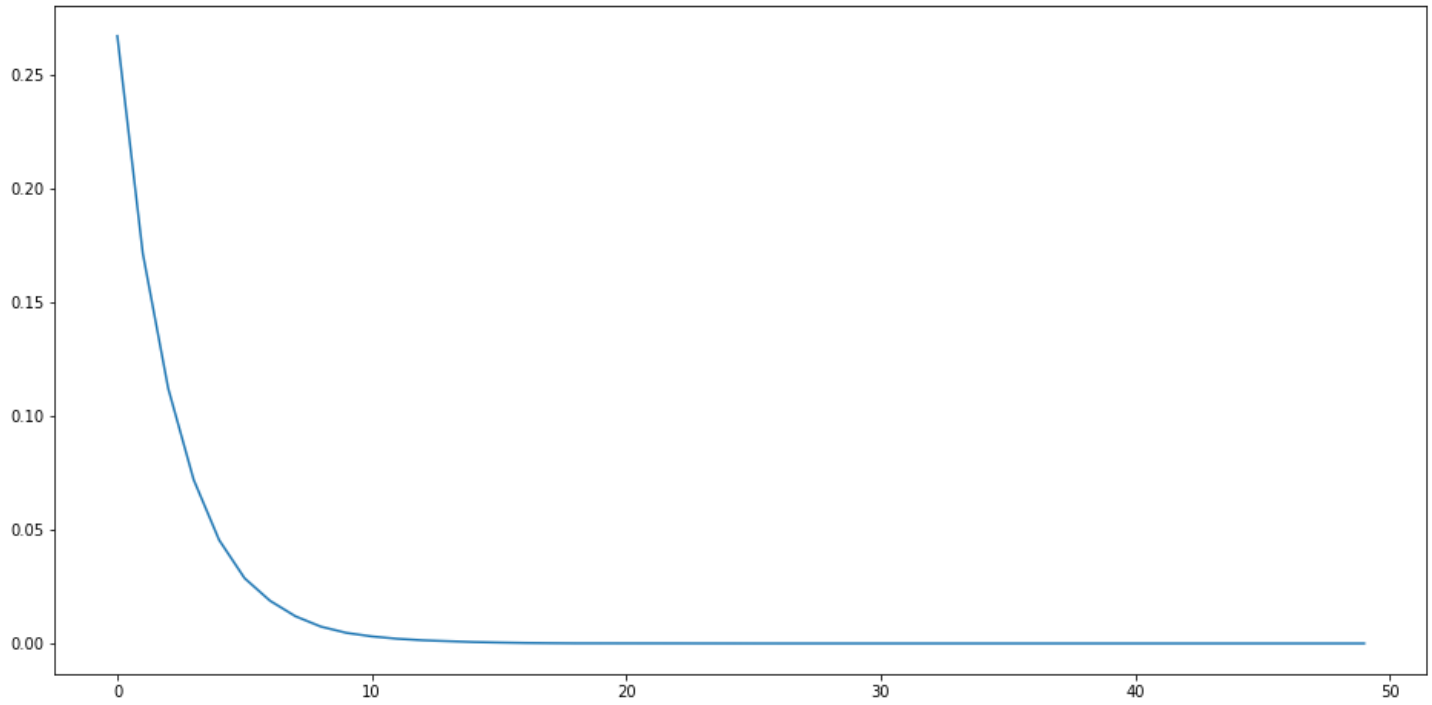
```
The mean is  497.228


The varience is  3576035.292016
```



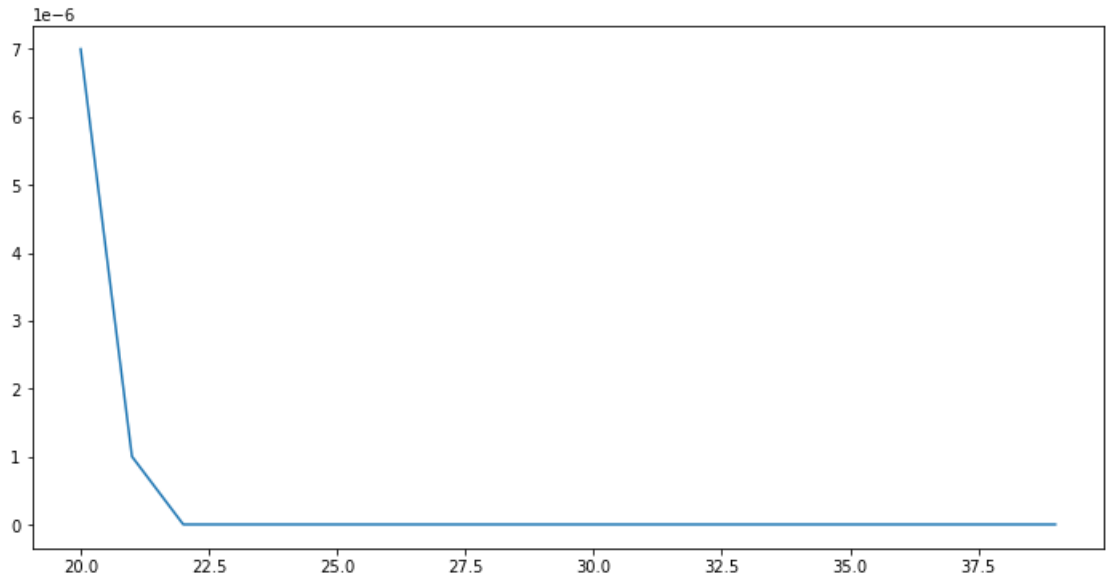
In [5]:

```
import random
import numpy as np
import matplotlib.pyplot as plt

n_t = 51
sum = 0
res = 0
P = np.zeros(n_t)
for i in range(0,1000000):
  n = random.random()
  if (0.18<n<0.58 and sum>0):
    sum = sum - 1
  if(n<0.3):
    sum = sum + 1
  res+=sum
  if sum < 51 : P[sum]+=1
x=0
P_new =P[1+x:50+1+x]/float(1000000)
print("\n\n",float(res)/1000000,"\n\n")

plt.figure(figsize=(16,8))
plt.plot(np.arange(0+x,50+x),P_new)
plt.show()
```

```
 2.090915
```



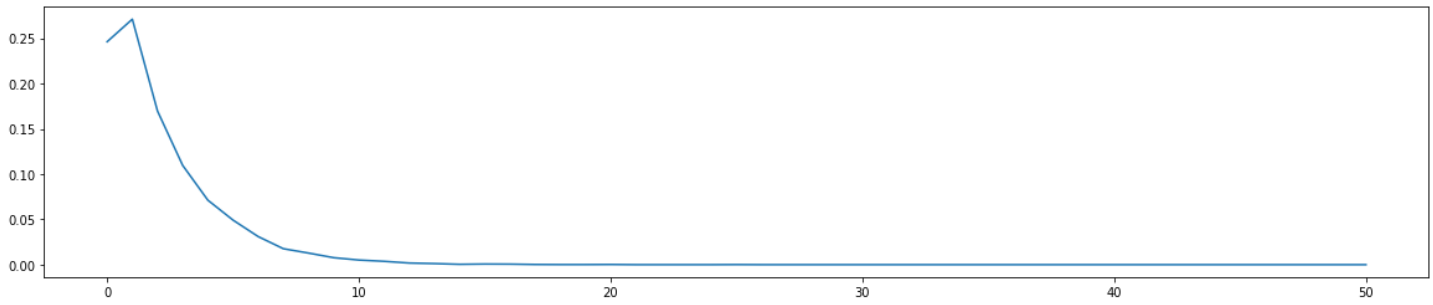
In [ ]:
```
x=20
P_new =P[1+x:20+1+x]/float(1000000)
plt.figure(figsize=(12,6))
plt.plot(np.arange(0+x,20+x),P_new)
plt.show()
```



In [1]:
```
import random
import numpy as np
import matplotlib.pyplot as plt

n_arr=[0]*51
res=0
for j in range(0,10000):
  sum = 0
  #if j % 500 == 0 : print(j)
  for i in range(0,100000):
    n = random.random()
    if (0.18<n<0.58 and sum>0):
      sum = sum - 1
    if(n<0.3):
      sum = sum + 1
  if sum < 51 : n_arr[sum]+=1
  res+=sum
for i in range(51):
  n_arr[i]=float(n_arr[i])/10000
```

```
print(float(res)/10000)
plt.figure(figsize=(20,4))
plt.plot(np.arange(0,len(n_arr)),n_arr)
plt.show()
```

```
2.1189
```



In [2]:

```
x=10
y=20
plt.figure(figsize=(16,8))
plt.plot(np.arange(0+x,len(n_arr[x:y])+x),n_arr[x:y])
plt.show()
```