Ques 1

Simulate 5000 sample of exponential with mean 5. Draw the histogram and the calculate the mean, maximum and minimum. (Use R and C/C++)

C++ code:-

```
#include<stdio.h>
#include<math.h>
#include<malloc.h>
int check(float z)
{
  int pos,i;
  float a[]={0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.0,100.0};
   for(i=0;i<21;i++)
    {
       if(z<=a[i]/2.5)
       {
          pos=i;
          break;
       }
    }
  return pos;
}

void main()
{

  float z,f[5000],mean=0;
  int i=0,freq[20]={0};
  long int y,x=123;
  while(i<5000)
   {
      y=(1597*x)%244944;
      z=(float)((float)y/244944);
      f[i]=z;
      i++;
      x=y;
   }

  for(i=0;i<5000;i++)
   {
      f[i]=-5*log(1-f[i]);
      mean=mean+f[i];
   }

   for(i=0;i<5000;i++)
   {
      freq[check(f[i])-1]++;
   }

   mean=mean/5000;

   float max=f[0],min=f[0];
```

```
48.
49.    for(i=0;i<5000;i++)
50.    {
51.       if(max<f[i])
52.         max=f[i];
53.       if(min>f[i])
54.         min=f[i];
55.
56.       printf("%f\n",f[i]);
57.    }
58.    for(i=0;i<20;i++)
59.     printf("%d %d\n",i,freq[i]);
60.
61.     printf("\n MEAN IS %f \n MAX IS %f \n MIN IS %f\n",mean,max,min);
62. }
```

R code:-

```
1.    func<-function()
2.   {
3.     mean<-5;
4.     x0<-123;
5.     a<-1597;
6.     m<-244944;
7.     y<-x0;
8.     u<-vector(len=5000);
9.     exp<-vector(len=5000);
10.
11.    for(i in 1:5000)
12.    {
13.      y<-((a*y)%%m);
14.      if(y<0)
15.        y<-y+m;
16.
17.      u[i]<-y/m;
18.      exp[i]<--mean*log(1-u[i]);
19.    }
20.    png("exp_R.png");
21.    hist(exp,breaks=50,col='black',plot=TRUE);
22.    dev.off();
23.
24.    cat("\nThe mean of values generated is ",mean(exp))
25.    cat("\nThe maximum is ",max(exp))
26.    cat("\nThe minimum is ",min(exp),"\n")
27. }
```

Output of Program:-

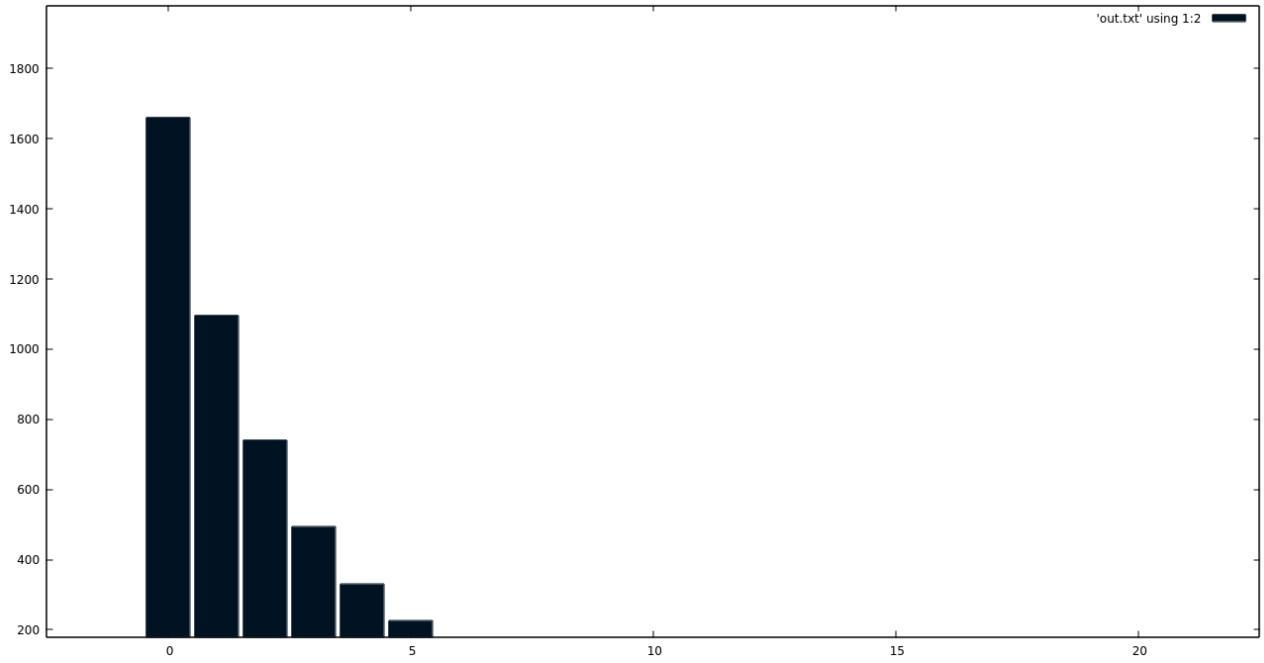
Expected mean is 5

Calulated from Program

MEAN IS 4.990046
MAX IS 37.744713
MIN IS 0.002511

Graph :-

Graph showing exponential distribution



Ques 2

Generate 5000 sample from Gamma with parameter n = 5 and λ = 5. Draw the histogram and the calculate the mean, maximum and minimum. (Use R and C/C++)

C++ Code:-

```
1.  #include<stdio.h>
2.  #include<math.h>
3.  #include<malloc.h>
4.  int check(float z)
5.  {
6.    int pos,i;
7.    float
    a[]={0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.
    0,100.0};
8.     for(i=0;i<21;i++)
9.      {
10.         if(z<=a[i]/25)
```

```
        {
          pos=i;
          break;
        }
     }
    return pos;
}

void main()
{

   float z,f[25000],f1[5000],mean=0;
   int i=0,j,freq[20]={0};
   long int y,x=123;
   while(i<25000)
    {
      y=(1597*x)%244944;
      z=(float)((float)y/244944);
      f[i]=z;
      i++;
      x=y;
    }

   for(i=0;i<25000;i++)
    {
      f[i]=-1*log(1-f[i])/5;

    }

    for(i=0;i<5000;i++)
    {
      f1[i]=f[i]+f[i+5000]+f[i+10000]+f[i+15000]+f[i+20000];

      mean=mean+f1[i];
   }
     for(i=0;i<5000;i++)
    {
      freq[check(f1[i])-1]++;
    }

    mean=mean/5000;

    float max=f1[0],min=f1[0];

    for(i=0;i<5000;i++)
    {
      if(max<f1[i])
        max=f1[i];
      if(min>f1[i])
        min=f1[i];

//     printf("%f\n",f1[i]);
    }
    for(i=0;i<20;i++)
     printf("%d %d\n",i,freq[i]);

     printf("\n MEAN IS %f \n MAX IS %f \n MIN IS %f\n",mean,max,min);
```

```
68. }
```

R code:-

```
1.  func<-function()
2.  {
3.          x_initial<-123
4.          a<-1579
5.          m<-244944
6.          b<-0
7.          l<-5
8.          mean<-1/l
9.
10.         u<-vector(length=5000)
11.         expn<-vector(length=5000)
12.         gamma<-vector(length=5000)
13.
14.         for(i in 1:5000)
15.          gamma[i]=0
16.
17.         while(x_initial<=100009)
18.         {
19.                 y=x_initial
20.                 for(j in 1:5000)
21.                 {
22.                         y<-((a*y+b)%%m)
23.                         if(y<0)
24.                          y<-y+m;
25.                         u[j]<-y/m
26.                         expn[j]<-(-mean)*log(1-u[j])
27.                         gamma[j]<-gamma[j]+expn[j]
28.                 }
29.                 x_initial<-x_initial+2
30.         }
31.
32.         png("gamma_R.png");
33.         hist(gamma, breaks=50, col="blue", plot=TRUE);
34.         dev.off();
35.
36.         cat("\nThe mean of values generated is ",mean(gamma))
37.         cat("\nThe maximum is ",max(gamma))
38.         cat("\nThe minimum is ",min(gamma),"\n")
39. }
```

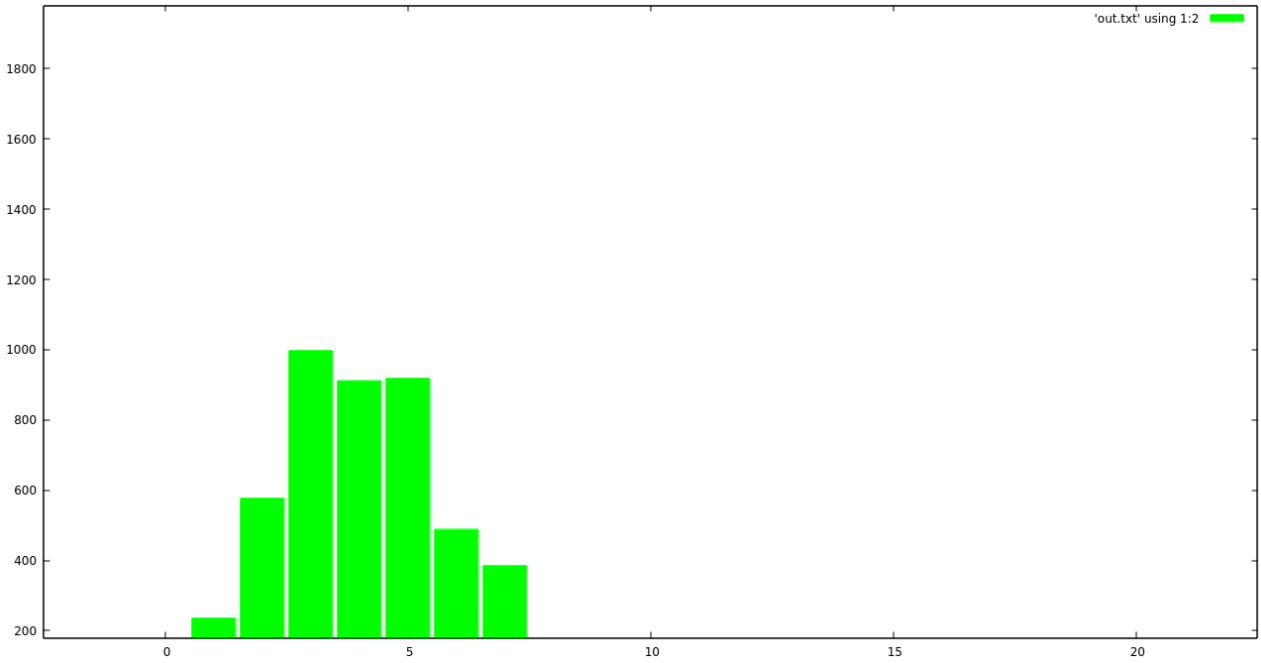
Output of Program:-

Expected Mean = 1
Calculated from Program

MEAN IS 1.000673
MAX IS 3.747443
MIN IS 0.059284

Graph of Program:-

Graph showing gamma distribution



Ques 3

Use the rejection method to generate from $f(x) = 20x(1 - x)^3$, $0 < x < 1$. (Use R)

C++ Code:-

```
#include<stdio.h>
#include<math.h>
int check(float z)
{
  int pos,i;
  float a[]={0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.0,100.0};
   for(i=0;i<21;i++)
    {
       if(z<=a[i]/100)
       {
         pos=i;
         break;
       }
    }
  return pos;
}

float f(float x)
{
  return (20*x*pow((1-x),3));
}

void main()
{
  int c=3;
```

```
26.  float z,f2[5000],f1[5000],g[5000],mean=0;
27.  int i=0,j,freq[20]={0};
28.  long int y,x=123;
29.  while(i<5000)
30.  {
31.    y=(1597*x)%244944;
32.    z=(float)((float)y/244944);
33.    f2[i]=z;
34.     i++;
35.    x=y;
36.  }
37.  i=0;
38.  x=34;
39.
40.  while(i<5000)
41.  {
42.    y=(1597*x)%244944;
43.    z=(float)((float)y/244944)*3;
44.    f1[i]=z;
45.
46.    i++;
47.    x=y;
48.  }
49.
50.
51.  for(i=0,j=0;i<5000;i++)
52.  {
53.
54.   if(3*f1[i]<=f(f2[i]))
55.    {
56.      g[j]=f2[i];
57.      j++;
58.    }
59.   }
60.  for(i=0;i<j;i++)
61.   {
62.    freq[check(g[i])-1]++;
63.   }
64.   for(i=0;i<20;i++)
65.    printf("%d %.2f %d\n",i,i*0.05,freq[i]);
66.
67. }
```

R Code:-

```
1.  Rejection<-function()
2.  {
3.          x<-vector(length=5000)
4.          x1<-1000001
5.          x2<-1000005
6.          i<-0;
7.          a<-16807
8.          b<-0
9.          m<-2^31-1
10.         while (i<=5000)
11.         {
12.                 y<-x1/m;
```

```
13.                 x1<-(a*x1+b)%%(m);
14.                 u<-x2/m;
15.                 x2<-((a*x2)+b)%%(m);
16.                 if (u<=((20*y*((1-y)^3)*64)/(27*5)))
17.                 {
18.                         x[i]<-y;
19.                         i<-i+1;
20.                 }
21.         }
22.
23.         png("rejection.png");
24.         hist(x, breaks=50, col="blue", plot=TRUE);
25.         dev.off();
26. }
```

Graph :-

Graph showing $20x(1-x)^3$ function