Name: Somya Khandelwal

Roll Number: 200123056

Department: Mathematics and Computing

Course: MA 323 - Monte Carlo Simulation

Lab: 11

In [20]:

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

In [21]:

```
def decimalToBinary(n):
  return bin(n).replace("0b", "")

def decimalToTernary (n):
  if n == 0:
    return '0'
  nums = []
  while n:
    n, r = divmod(n, 3)
    nums.append(str(r))
  return ''.join(reversed(nums))
```

Question 1

Generated the 25 values of the Van der Corupt Sequence along with the graph using:

$$\phi_b(i) := \sum_{k=0}^{\infty} d_k b^{-k-1}.$$

The base-b Van Der Corput sequence is given by

$$x_i := \phi_b(i).$$

In [22]:

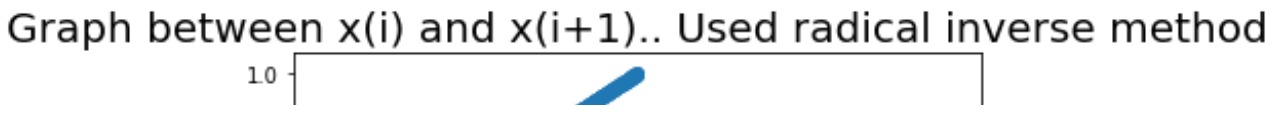
```
print("First 25 values of the Van der Corupt sequence, using the radical inverse
```

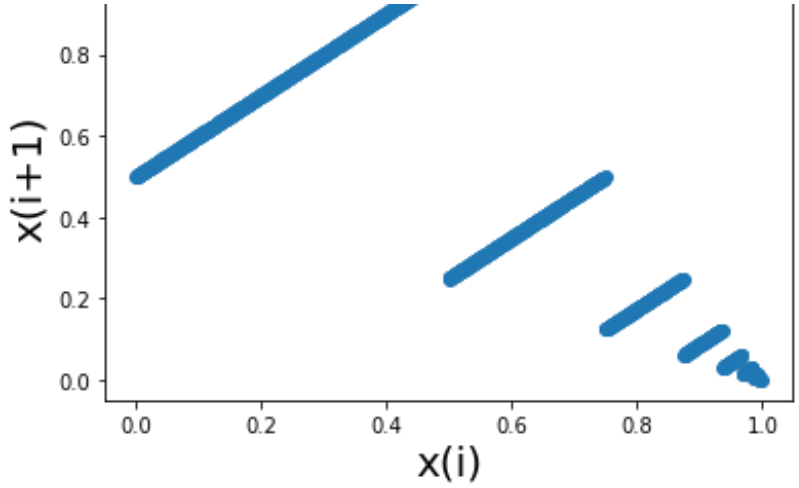
```
print("First 25 values of the Van der Corupt sequence, using the radical inverse
function are:")
for i in range(1, 26):
  s= decimalToBinary(i)
  value= 0
  for j in range(0, len(s)):
    if s[len(s)-j-1]=='1':
      value+= 1.00/pow(2, j+1)
  print(value)
```

```
First 25 values of the Van der Corupt sequence, using the radical inverse function
are:
0.5
0.25
0.75
0.125
0.625
0.375
0.875
0.0625
0.5625
0.3125
0.8125
0.1875
0.6875
0.4375
0.9375
0.03125
0.53125
0.28125
0.78125
0.15625
0.65625
0.40625
0.90625
0.09375
0.59375
```

In [23]:

```
Values= []
for i in range(0, 100000):
  s= decimalToBinary(i)
  value= 0
  for j in range(0, len(s)):
    if s[len(s)-j-1]=='1':
      value+= 1.00/pow(2, j+1)
  Values.append(value)
X= []
Y= []
for i in range(0, 999):
  X.append(Values[i])
for i in range(1, 1000):
  Y.append(Values[i])
plt.scatter(X,Y)
plt.xlabel("x(i)",size= 20)
plt.ylabel("x(i+1)",size= 20)
plt.title("Graph between x(i) and x(i+1).. Used radical inverse method",size= 20)
plt.show()
```





We observe :
1) All values generated were >= 0 and < 1.

2) We can see that this linear trend resembles the graph of LCG.

So, we may think, it might resemble U[0,1]

We will use the following LCG :
a=1597

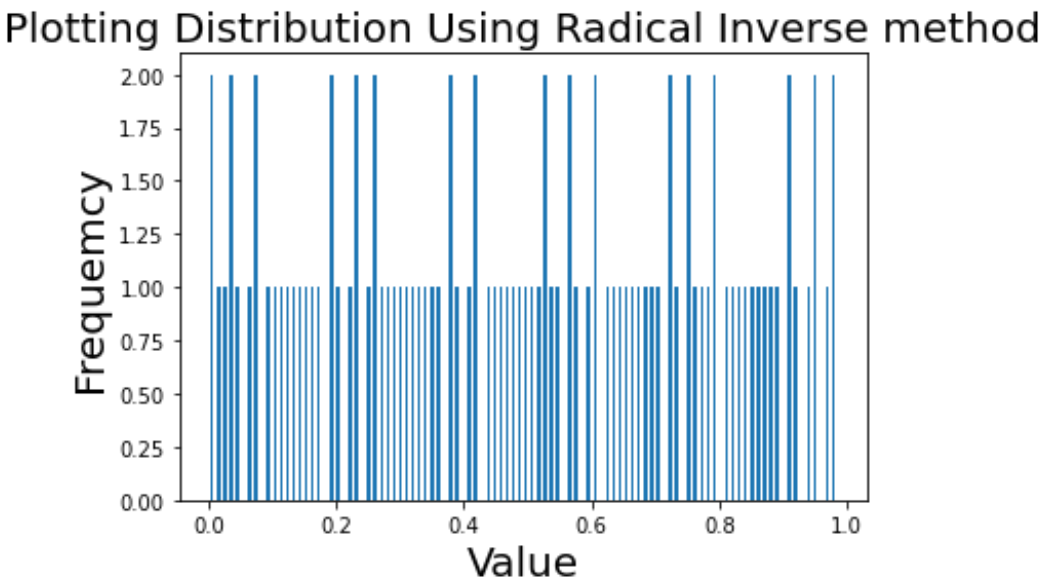
b=51749

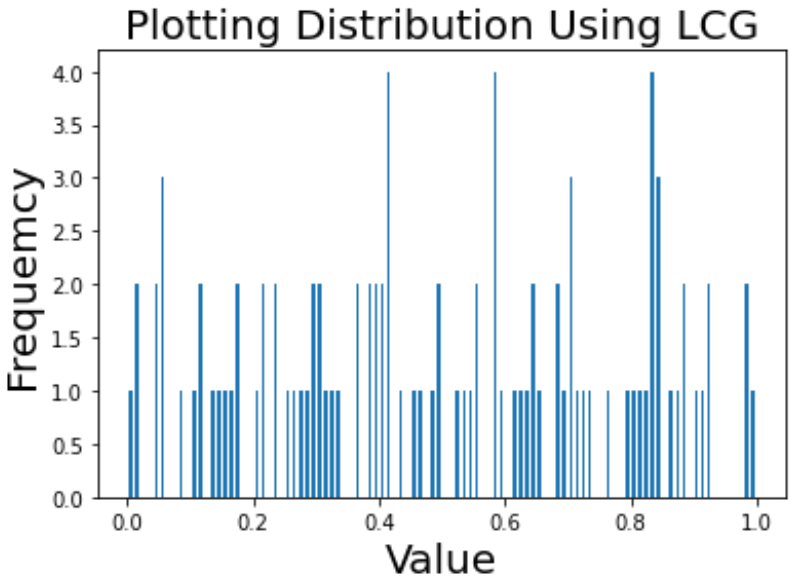
m=244944

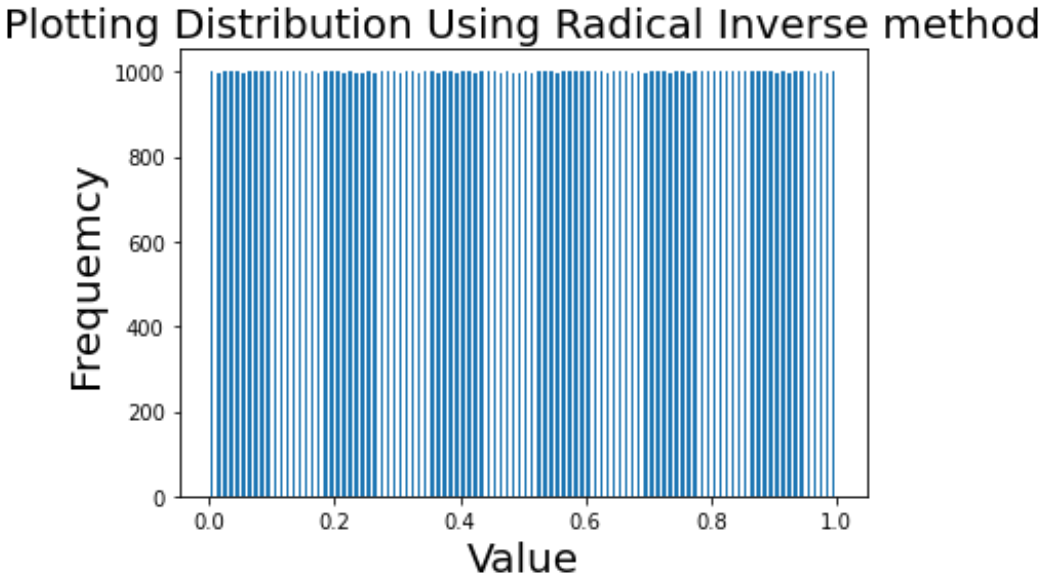
In [24]:

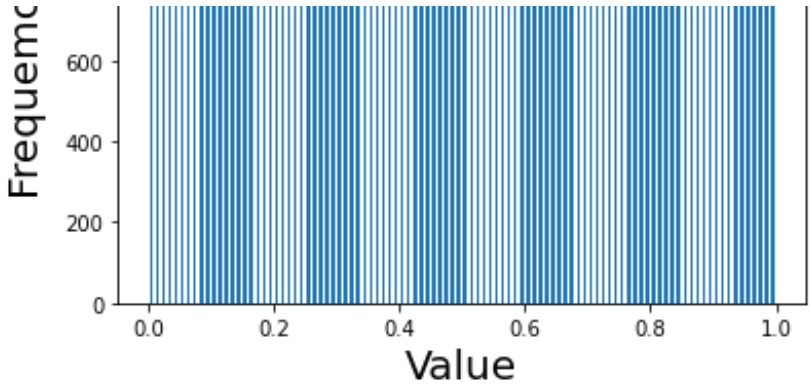
```
N= [100, 100000]
for n in N:
    Radical= []
    LCG= []
    for i in range(0, n):
        Radical.append(Values[i])
    m= 244944
    a= 1597
    b= 51749
    x0= 1
    for i in range(0, n):
        x= (a*x0 + b)%m
        LCG.append((x*1.000)/(m*1.000))
        x0= x
    plt.hist(Radical, bins = 100, rwidth = 0.5)
    plt.xlabel('Value',size= 20)
    plt.ylabel('Frequemcy',size= 20)
    plt.title('Plotting Distribution Using Radical Inverse method',size= 20)
    # plt.legend()
    plt.show()

    plt.hist(LCG, bins = 100, rwidth = 0.5)
    plt.xlabel('Value',size= 20)
    plt.ylabel('Frequemcy',size= 20)
    plt.title('Plotting Distribution Using LCG',size= 20)
```

```
    # plt.legend()
    plt.show()
```









We observe that :
1) As we increase n, we get a more uniform distribution.

2) Radical Inverse method has given better distribution for larger n

Question 2

Let $p_1, p_2, \ldots, p_m$ be a pairwise prime integers. The Halton sequence in m dimension is defined as follows:

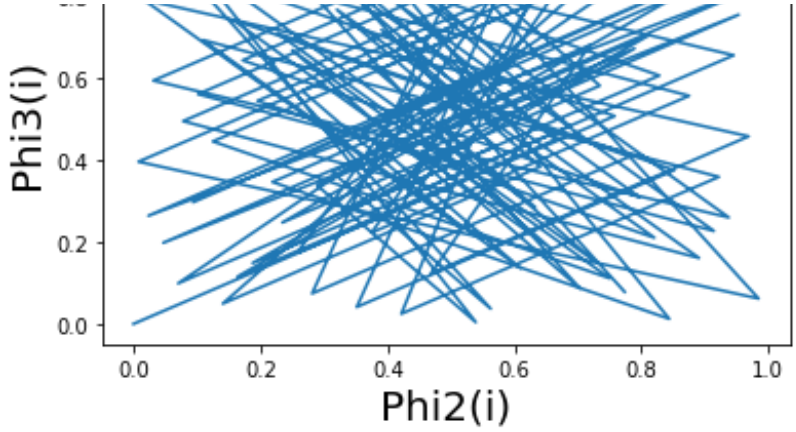
$$\boldsymbol{x}_i := (\phi_{p_1}(i), \phi_{p_2}(i), \ldots, \phi_{p_m}(i)), i = 1, 2, \ldots$$

Usually one takes $p_1, p_2, \ldots, p_m$ to the first m prime numbers.

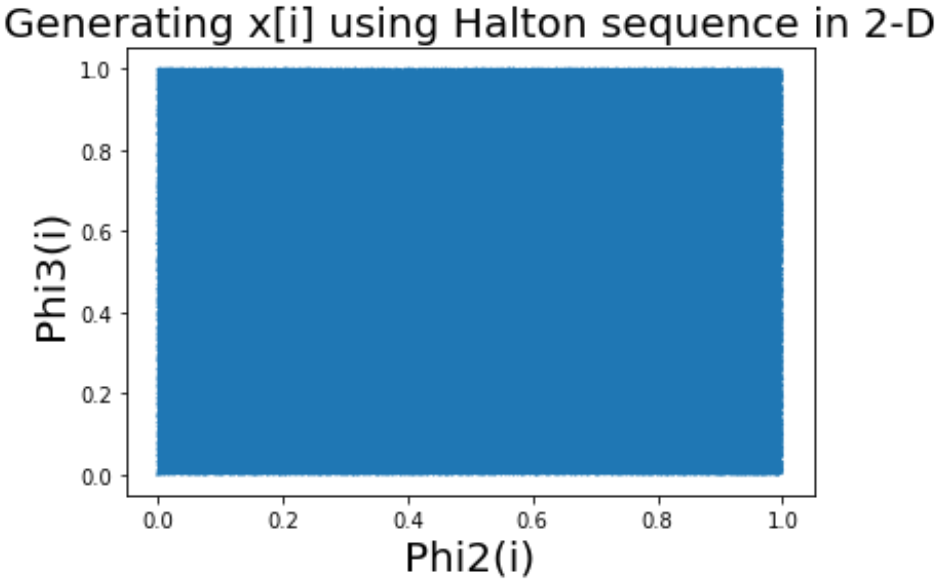
In [25]:

```
ValuesInTernary= []
plt.rcParams['agg.path.chunksize'] = 10000
for i in range(0, 100000):
    s= decimalToTernary(i)
    value= 0
    for j in range(0, len(s)):
        if s[len(s)-j-1]=='1':
            value+= 1.00/pow(3, j+1)
        if s[len(s)-j-1]=='2':
            value+= 2.00/pow(3, j+1)
    ValuesInTernary.append(value)

for n in N:
    XAxis= []
    YAxis= []
    for i in range(0, n):
        XAxis.append(Values[i])
        YAxis.append(ValuesInTernary[i])
    print("For n=",n)
    plt.plot(XAxis,YAxis)
    plt.xlabel("Phi2(i)",size= 20)
    plt.ylabel("Phi3(i)",size= 20)
    plt.title("Generating x[i] using Halton sequence in 2-D",size= 20)
    plt.show()
```

```
For n= 100
```



For n= 100000



We observe :

1) The above graphs of phi(2) vs phi(3) are uniformly distributed, and this this trend was expected, as 2 and 3 are relatively prime

2) The values of both phi(2) and phi(3) are in the range of [0,1)