ANT LAB Assignment 07 Random Number

Random Numbers: The Security of many cryptographic systems depends upon the generation of unpredictable quantities. Any predictability in the value of the quantities leads to a weakness in the entire system. It will be seen that true random numbers are very hard to produce accurately and require some physical phenomena which is not always practical. Failing the use of a true random source, a number of so called pseudorandom sources have been developed. These work much better in practical environments.

What are these basic criteria of random numbers?

1. **Uniform distribution:** The distribution of the numbers should be uniform within a specified range and should satisfy the statistical tests for randomness such as lack of predictability and of correlations among neighboring numbers.
2. **Independence:** The calculation should produce a large number of unique numbers before repeating the cycle.
3. The calculation should be very fast.

PROBLEM 1 :

In [ ]:

```c
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void main()
{
        int i,j,count,N=1000;
        int r[N],freq[N];
        // part a
        int a=5,r0=1,m=37;      // given values
        for(i=0;i<N;i++) {
                r[0]=r0;
                r[i+1]=(a*r[i])%m;
                freq[i]=-1;
        }
        // part b
        for (i=0;i<N;i++) {
                count=1;
                for(j=i+1;j<N;j++) {
                if(r[i]==r[j]) {
                        count++;
                        freq[j]=0;
                }
        }
        if(freq[i]!=0) {
                freq[i]=count;
                }
        }
        FILE*fp=NULL;
        fp=fopen("1a.txt","w");
        // stroing the frequecy distribution
        for (i=0;i<N;i++) {
                if(freq[i]!= 0) {
                        fprintf(fp,"%d\t%d\n",r[i],freq[i]);
```

```
                }
        }
        //Correlation Checks
    FILE*fp1=NULL;
    fp1=fopen("1b.txt","w");
    for(i=0;i<N-1;++i) {
        fprintf(fp1,"%d\t%d\n",r[i],r[i+1]);
    }
        // part c
        for(i=0;i<N;i++) {
                for(j=i+1;j<N;j++) {
                if (r[i]==r[j]) {
                        printf("The periodicity of the random no: %d\n",j);
                        exit(0);
                }
                }
        }
}
```

Do the random numbers generated using the above algorithm satisfy the “basic criteria” of random numbers.

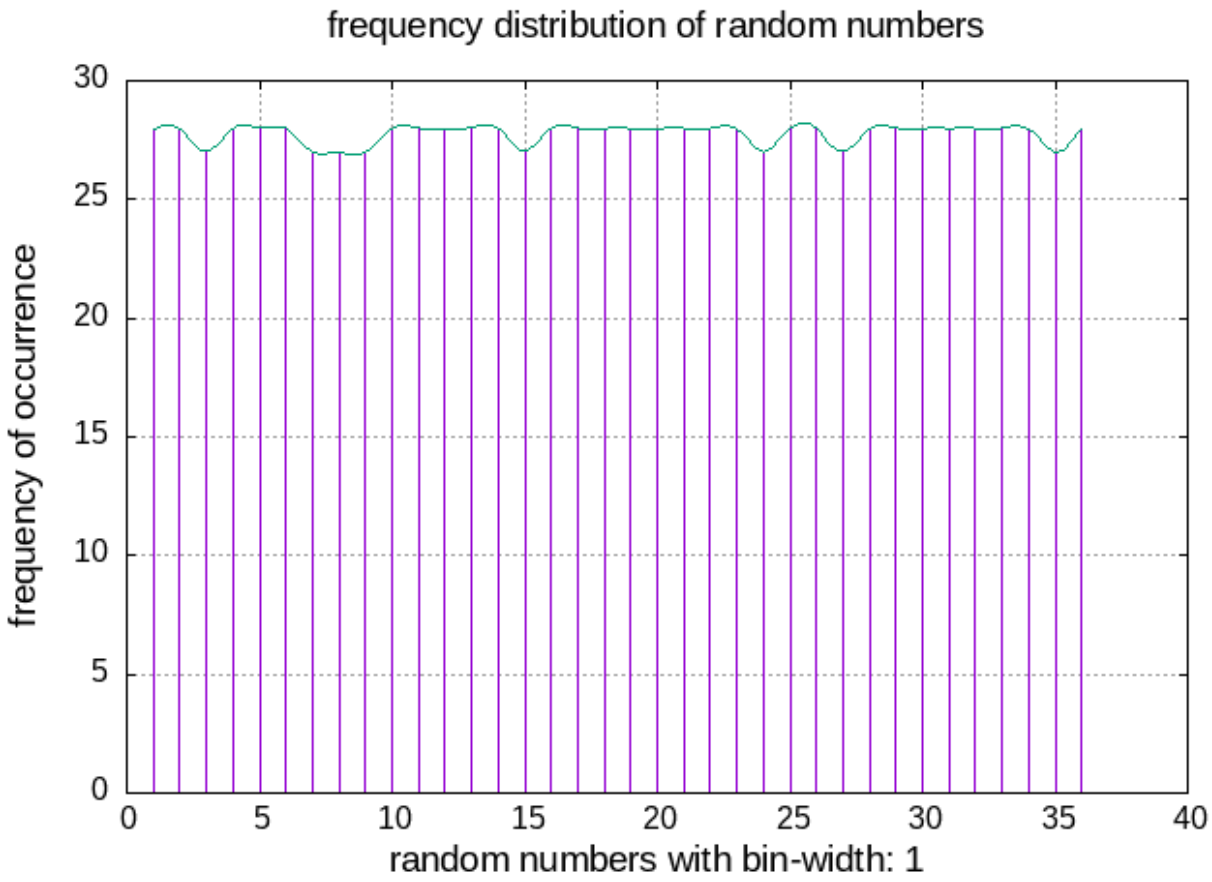
After generating the random numbers it appears that the generated random numbers by the given values of a and m are repeaating periodically after 36. Hence We can say it is good algorithm if we only want random numbers till 36.

Figure out the values of the “a” and “m” to generate 100,000 random numbers which satisfy the basic criteria of random number generation.

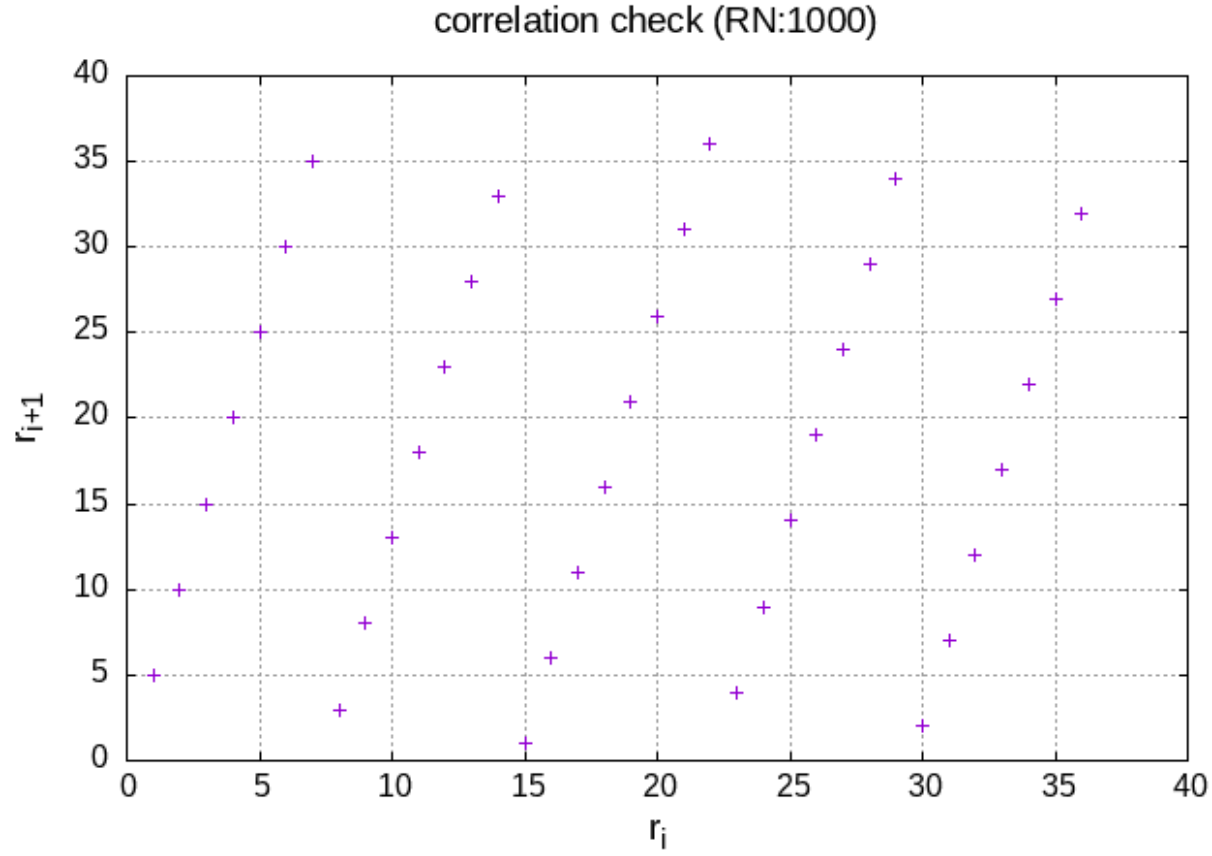
After searching over internet I found the values for a=16807 and m=2147483647 which was suggested by Par and Miller. They spent over 30 years surveying a large number of random number generators. The Phase spaces for the random numbers generated after plugin these values in the formula are also plotted in later page.

OUTPUT:

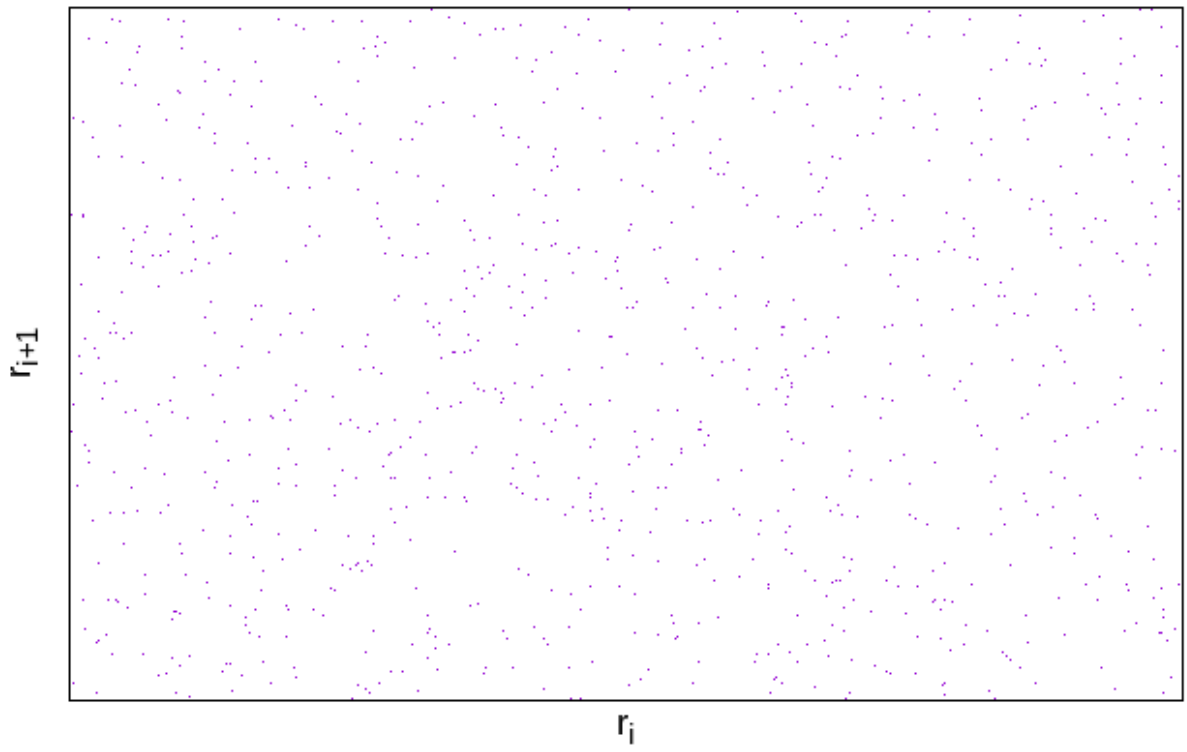
The periodicity of the random no: 36



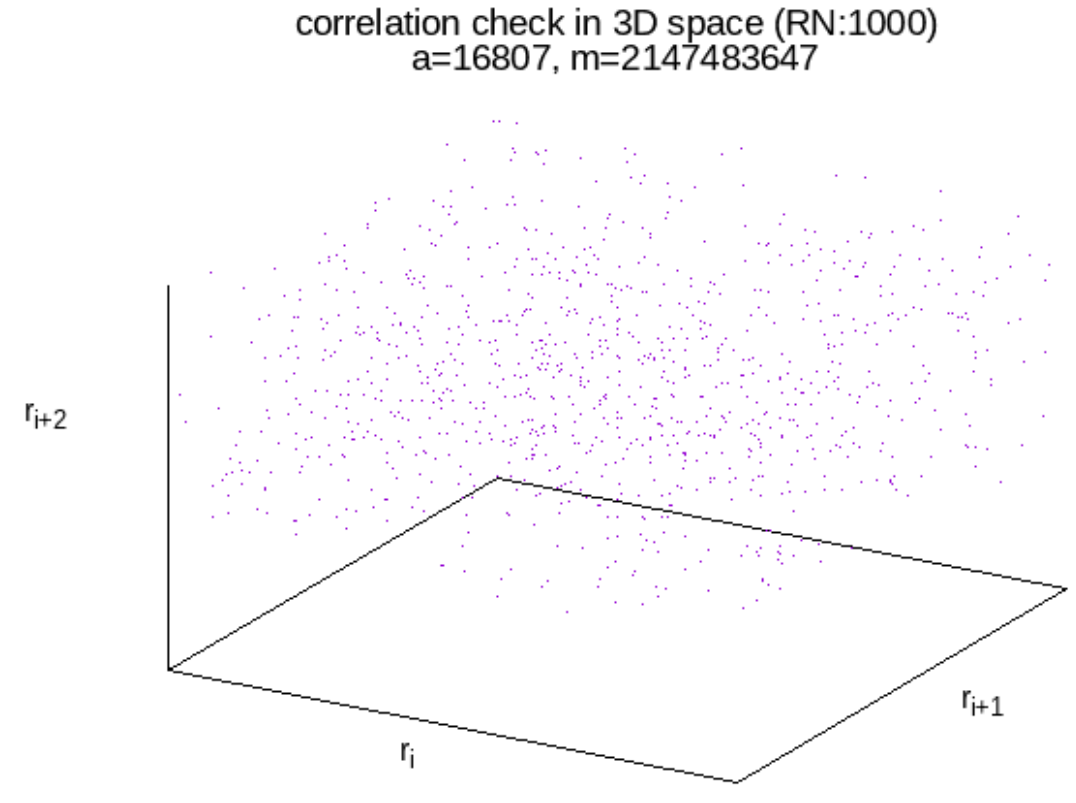
Sun Mar 21 14:35:44 2021



Sun Mar 21 14:35:04 2021



Sun Mar 21 15:06:33 2021



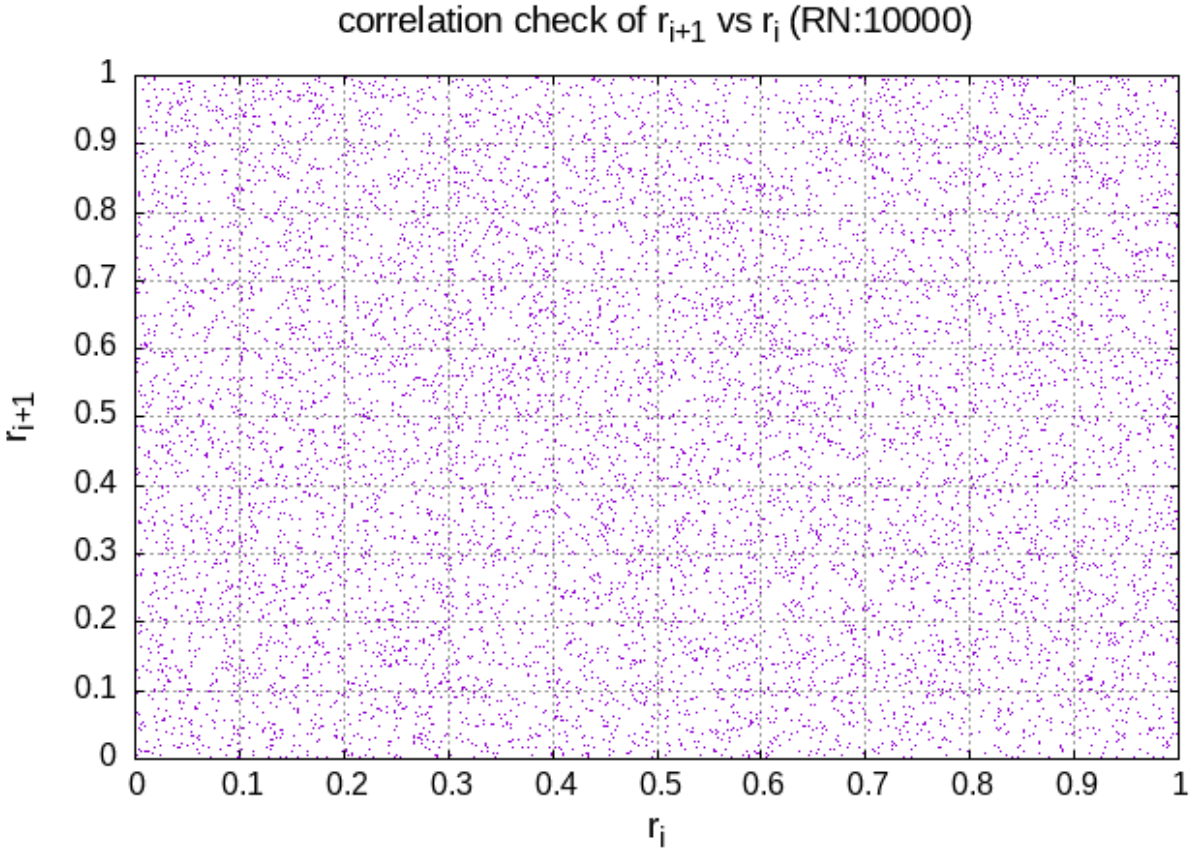
Sun Mar 21 15:06:56 2021

PROBLEM 2 :

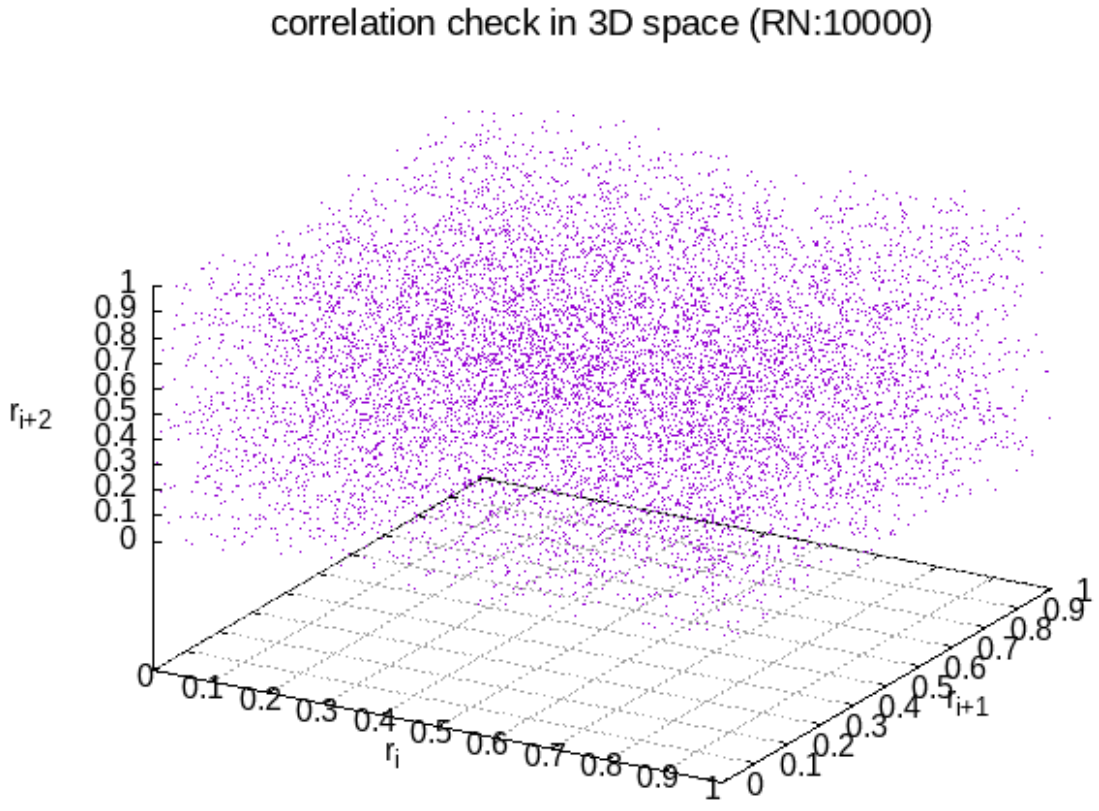
1. **`rand()` function:** The rand() function in the C programming language is used to generate a random number. The return type is of rand() function is an integer. It generates values in the range of 0 to RAND_MAX.
2. **`srand()` function:** The srand() function is used to set the starting value for the series of random integers. You can use the srand() to set the seed of the rand function to a different starting point.
3. **Generating random numbers between (0,1):** The value returned by rand() is in the range of 0 to RAND_MAX. When its divided by RAND_MAX, the result is in the range of 0.0 to 1.0. We can manipulate the real random numbers to represent values in any range. For example, multiply the values by 100 to have the code generate floating point numbers between 0.0 and 100.0
4. **Generating random numbers within a range (a,b):** To generate we manipulate using this formula: *number = (rand() % (b - a + 1)) + a*.

In [ ]:

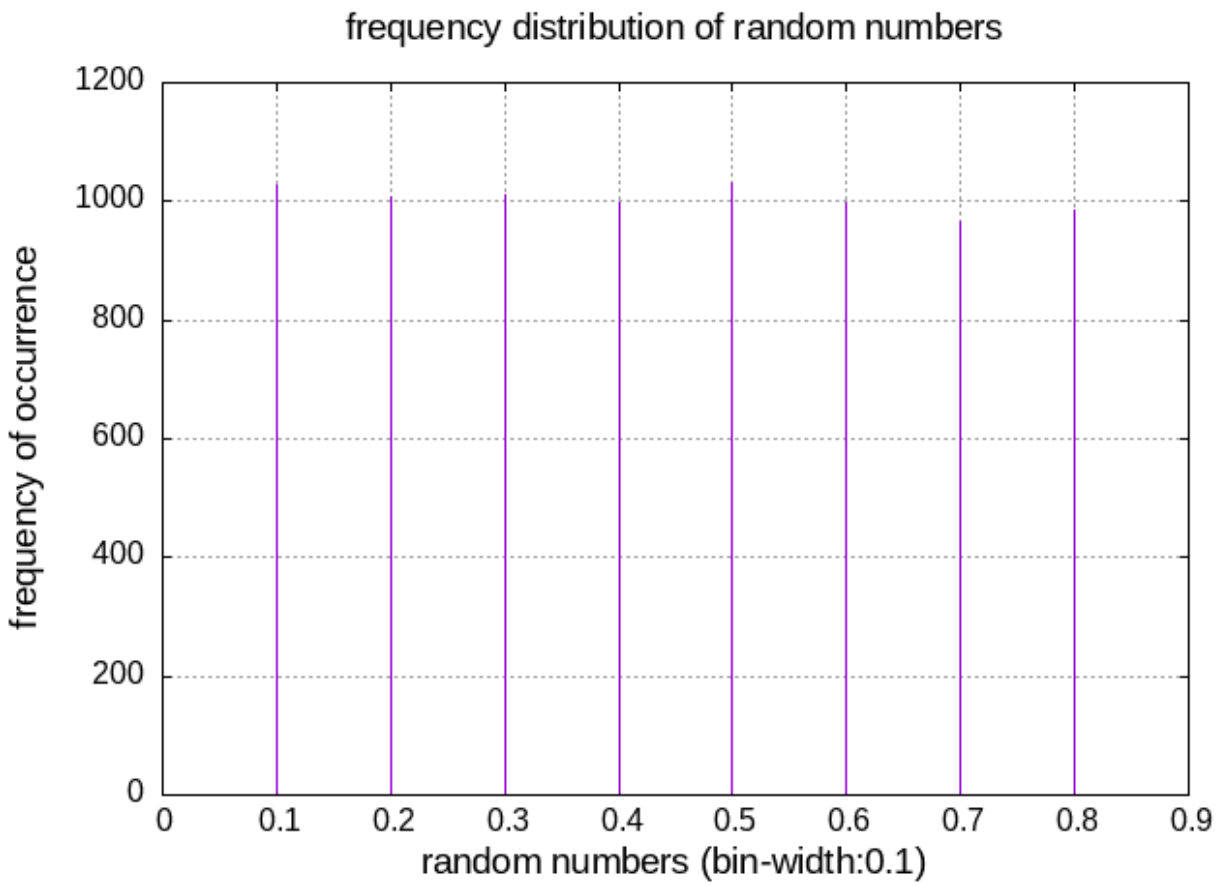
```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i,j,N=10000;
    double rn[N];
    //srand(time(0));
    for(i=0;i<N;++i) {
        rn[i]=((double)rand()/RAND_MAX);
    }
    //frequncy distribution within bin width
    double h=0.001; //width of interval
    int bin=1000;   //1000 intervals of width 0.001
    int freq[bin];
    for(j=0;j<bin;++j) {
        freq[j]=0;
        for(i=0;i<N;i++) {
            //frequncy of RN within bin width
            if((rn[i]>=j*h)&&(rn[i]<(j+1)*h)) {
                freq[j]++;
            }
        }
    }
    // stroing frequncy distribution
    FILE*fp=NULL;
    fp=fopen("2c1.txt","w");
    for(j=0;j<bin;++j) {
        fprintf(fp,"%lf\t%d\n",j*h,freq[j]);
    }
    //Correlation Checks
    FILE*fp1=NULL;
    fp1=fopen("2c2.txt","w");
    for(i=0;i<N-2;++i) {
        fprintf(fp1,"%lf\t%lf\t%lf\n",rn[i],rn[i+1],rn[i+2]);
    }
}
```



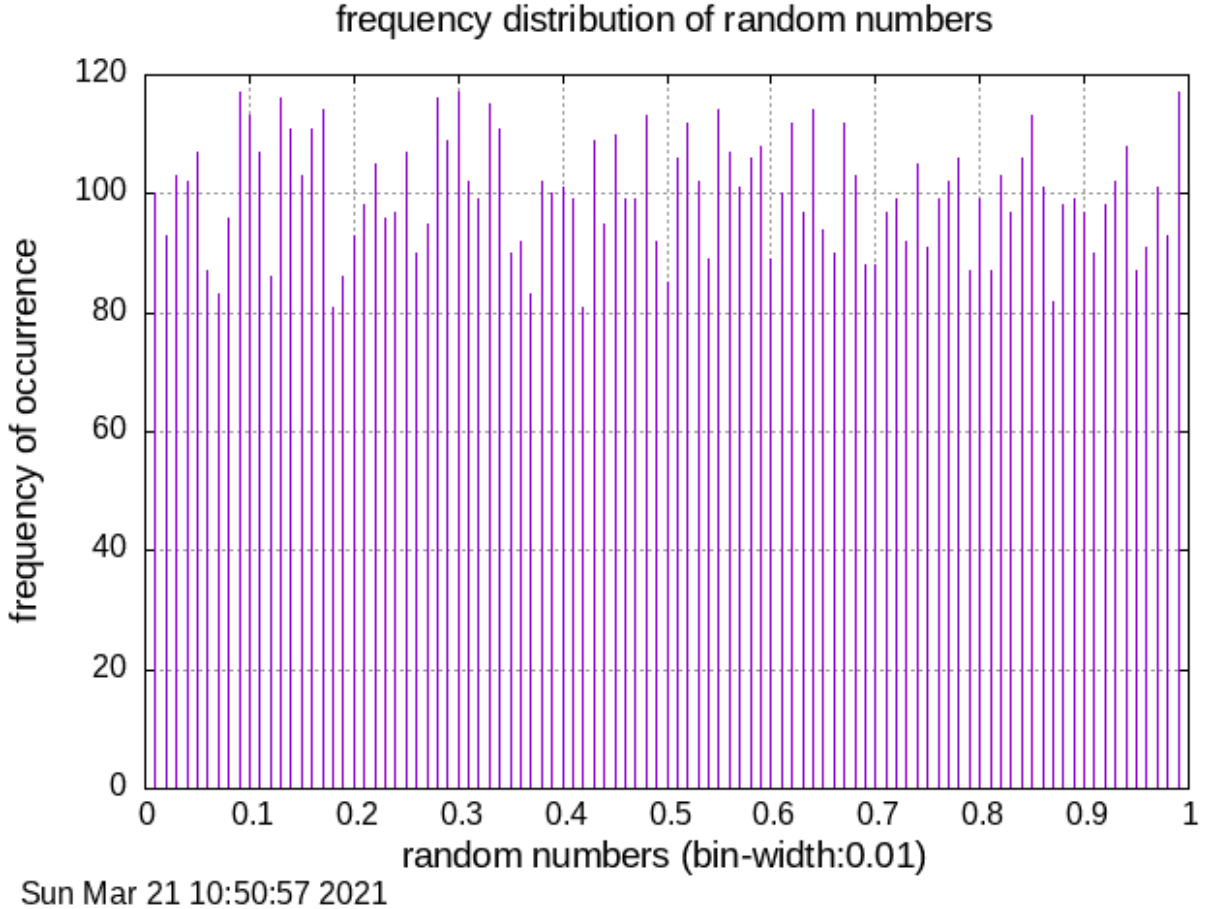
Sun Mar 21 10:41:34 2021



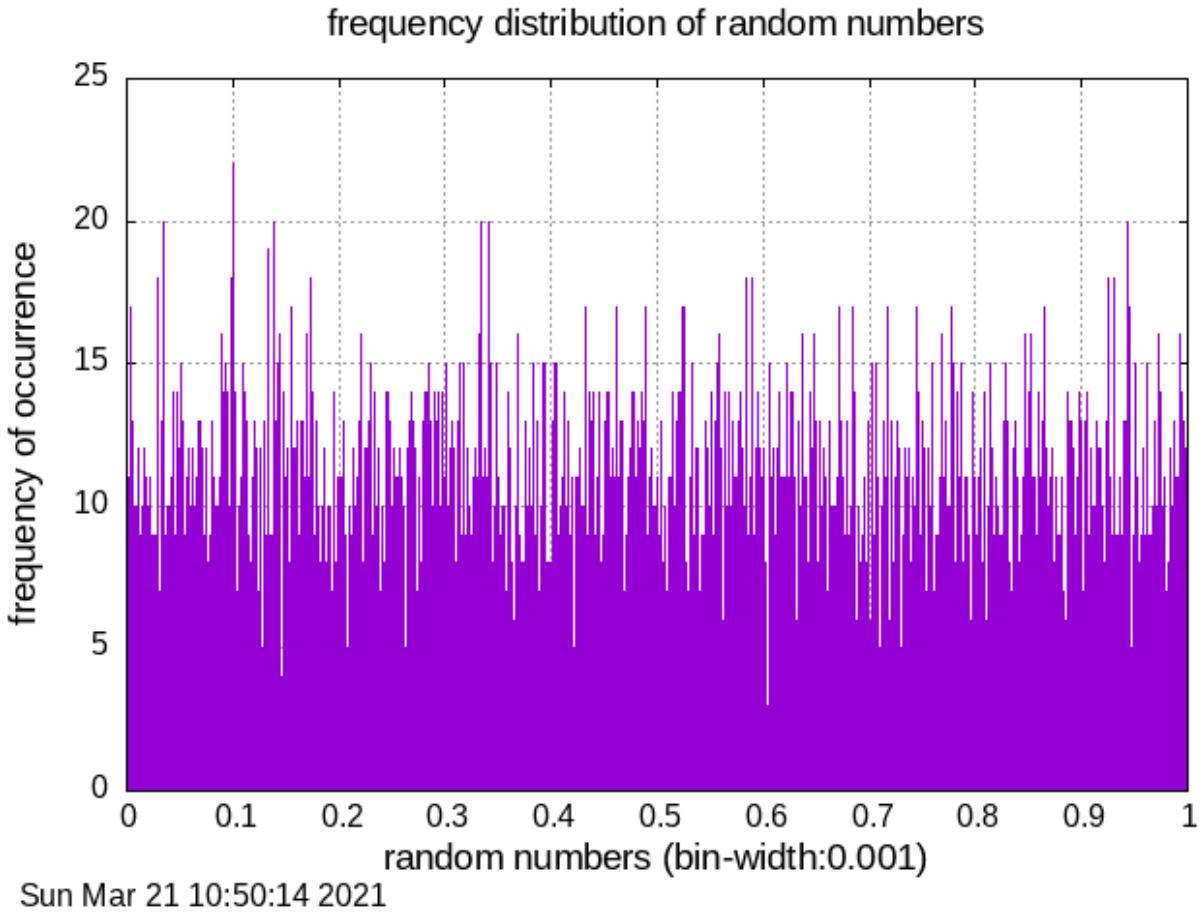
Sun Mar 21 10:41:50 2021



Sun Mar 21 10:51:14 2021



Sun Mar 21 10:50:57 2021

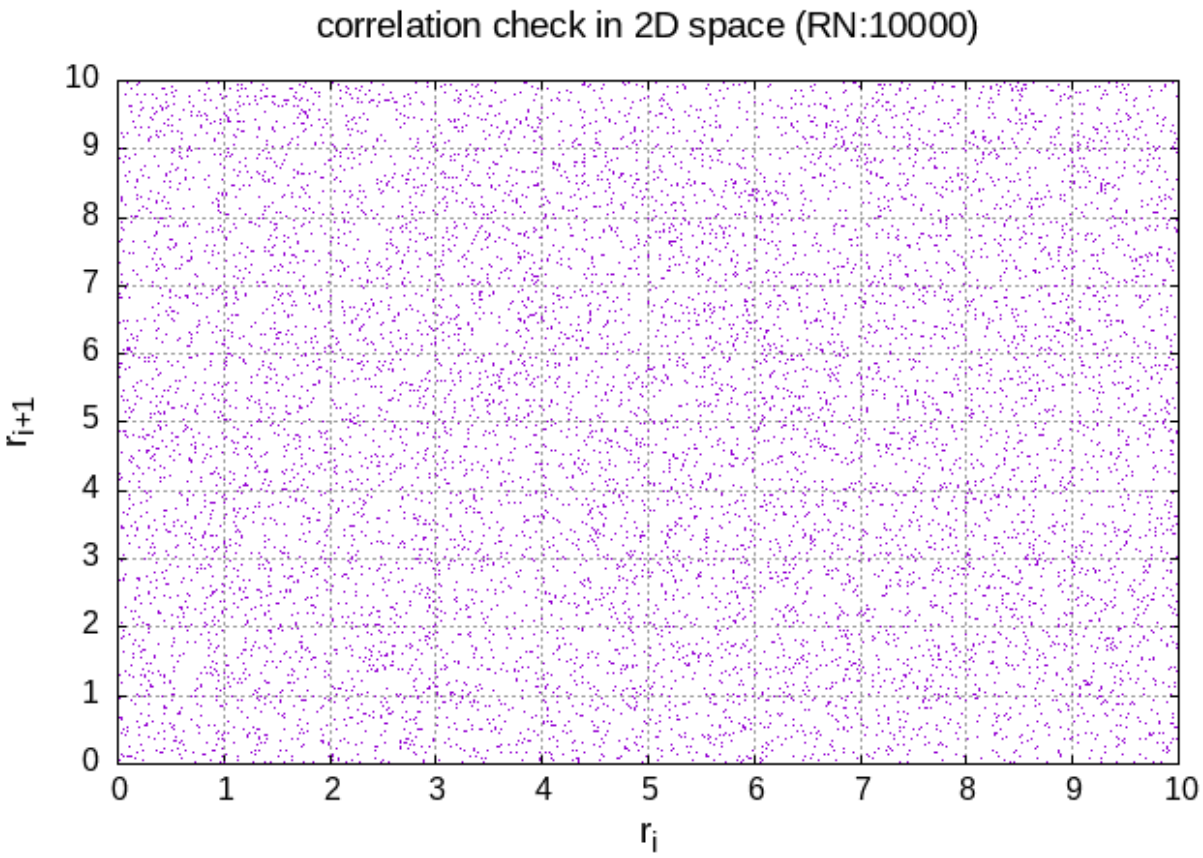


PROBLEM 3 : Case I

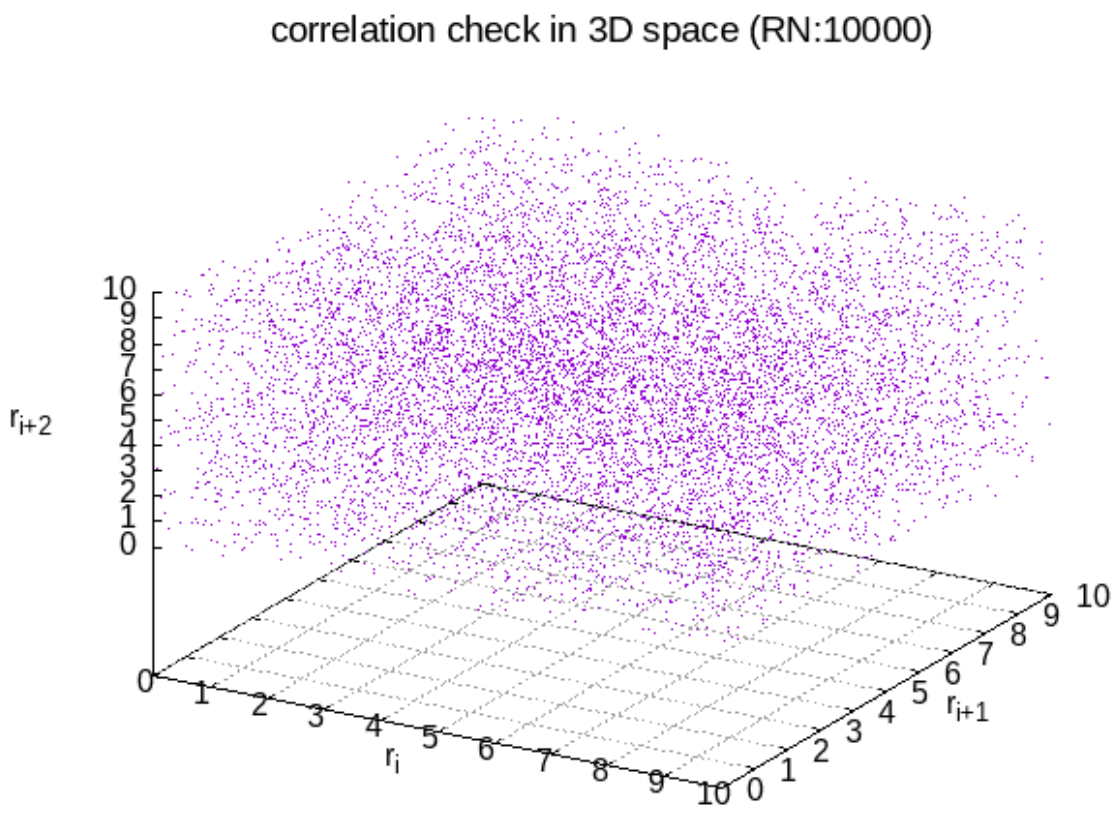
In [ ]:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i,j,N=10000,max=10;
    double M[N];
    //srand(time(0));
    for(i=0;i<N;++i) {
        M[i]=((double)rand()/RAND_MAX)*max;
    }
    //frequncy distribution within bin width
    double h=0.1; //width of interval
    int bin=10*max; //10 intervals of width 0.1 in [0:1]
    int freq[bin];
    for(j=0;j<bin;++j) {
        freq[j]=0;
        for(i=0;i<N;i++) {
            //frequncy of RN within bin width
            if((M[i]>=j*h)&&(M[i]<(j+1)*h)) {
                freq[j]++;
            }
        }
    }
    // stroing frequncy distribution
    FILE*fp=NULL;
    fp=fopen("3c1a.txt","w");
```

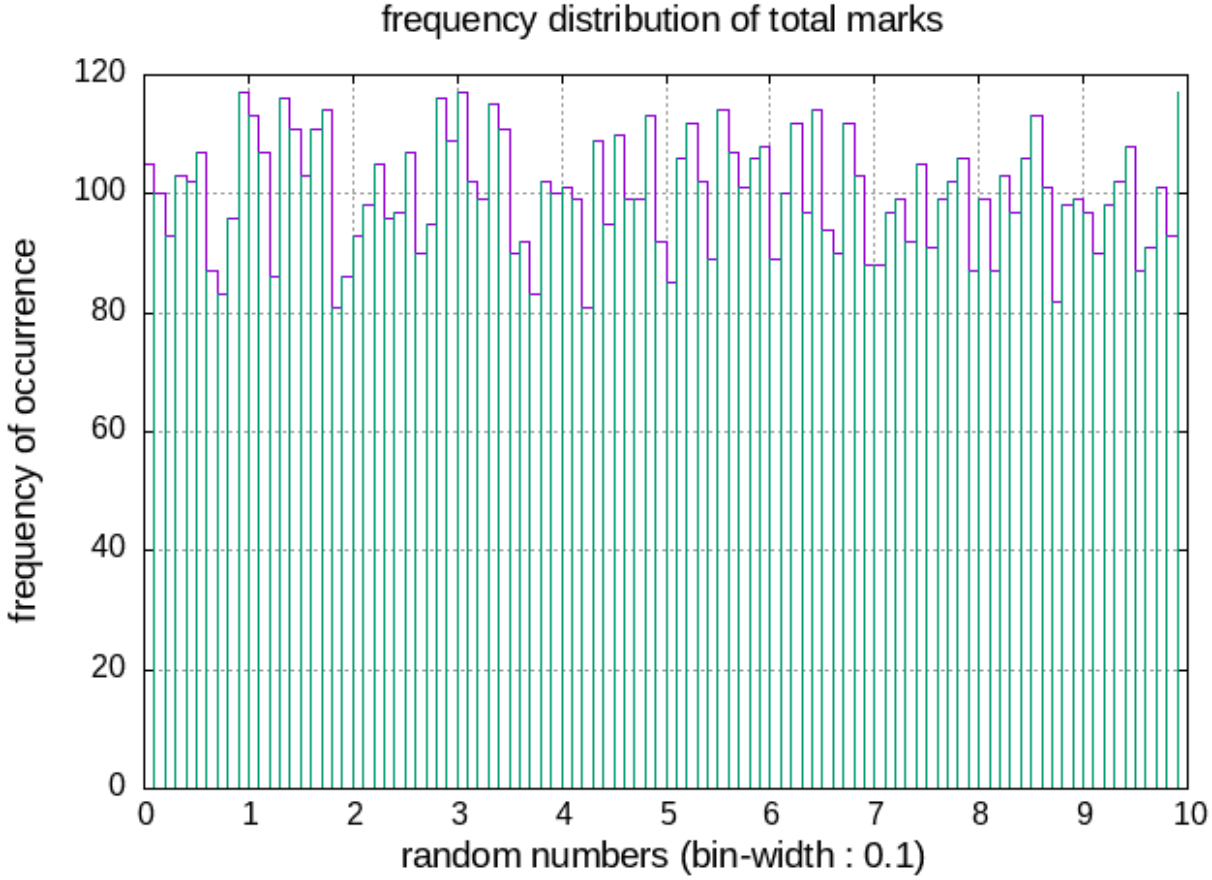
```
    for(j=0;j<bin;++j) {
        fprintf(fp,"%lf\t%d\n",j*h,freq[j]);
    }
    //Correlation Checks
    FILE*fp1=NULL;
    fp1=fopen("3c1b.txt","w");
    for(i=0;i<N-2;++i) {
        fprintf(fp1,"%lf\t%lf\t%lf\n",M[i],M[i+1],M[i+2]);
    }
}
```



Sun Mar 21 12:08:53 2021



Sun Mar 21 12:09:30 2021

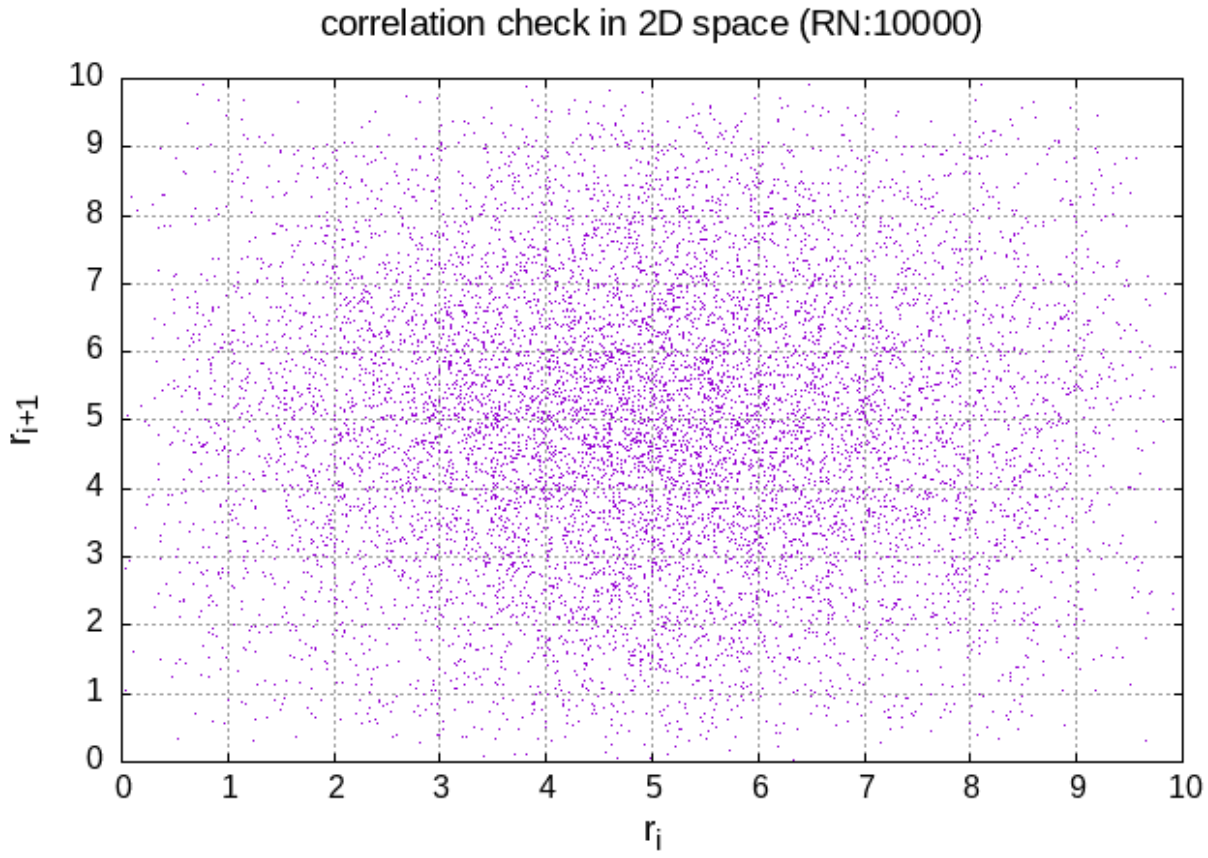


Sun Mar 21 12:15:16 2021

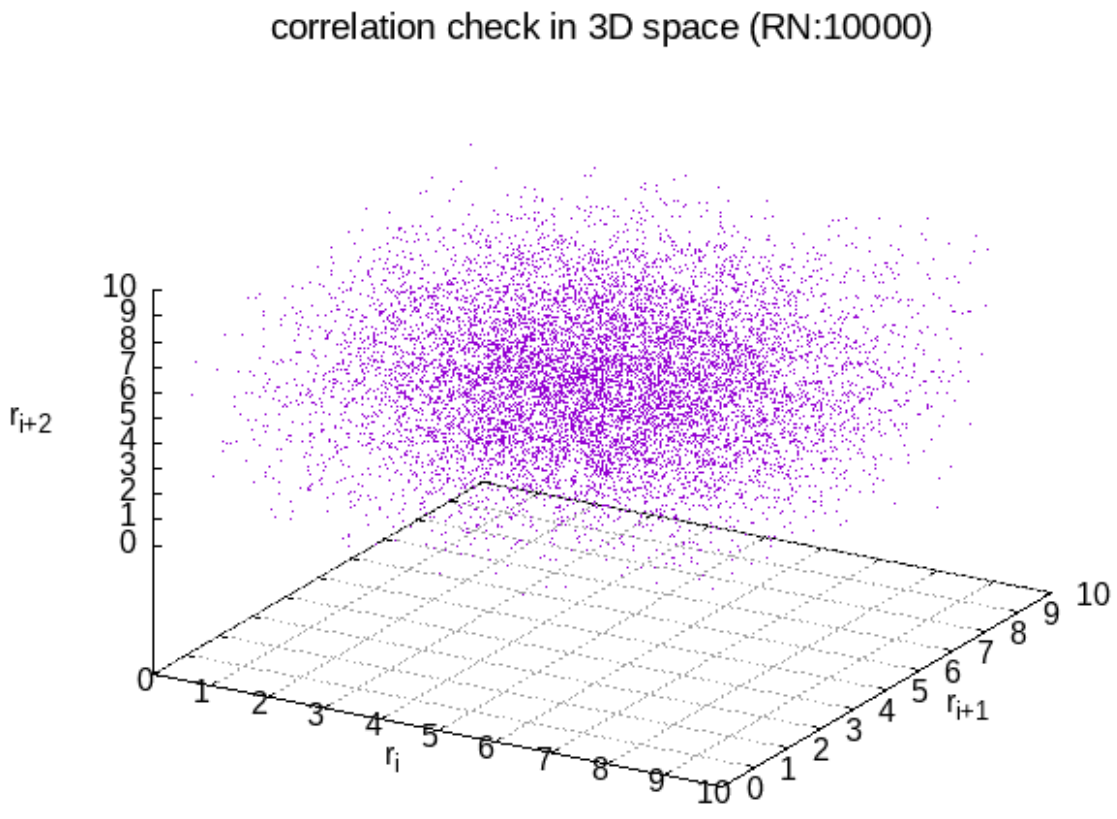
PROBLEM 3 : Case II

In [ ]:

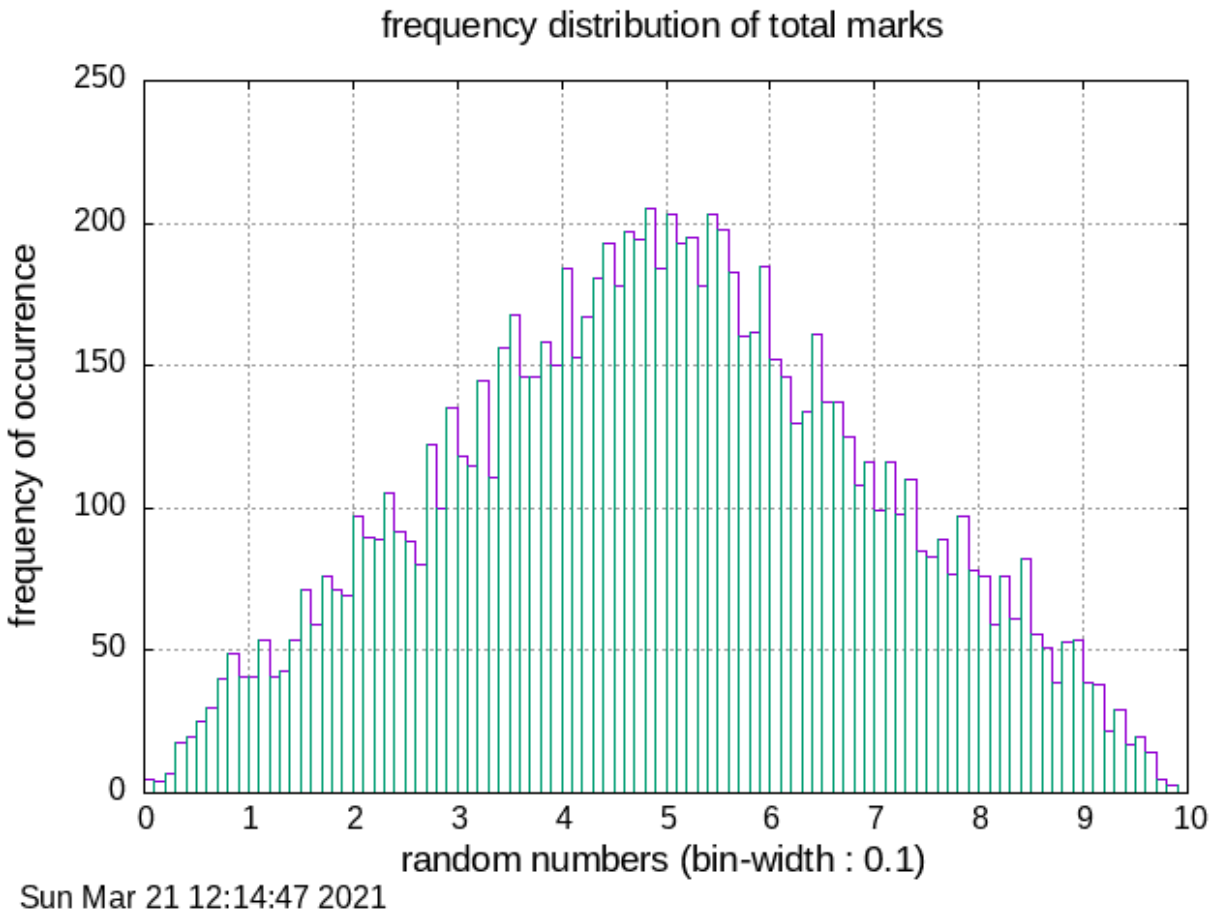
```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i,j,N=10000,max=5,sub=2;//subject
    double a[N],b[N],M[N];
    //srand(time(0));
    for(i=0;i<N;++i) {
        a[i]=((double)rand()/RAND_MAX)*max;
        b[i]=((double)rand()/RAND_MAX)*max;
        M[i]=a[i]+b[i];
    }
    //frequncy distribution within bin width
    double h=0.1; //width of interval
    int bin=10*sub*max; //10 intervals of width 0.1 in [0:1]
    int freq[bin];
    for(j=0;j<bin;++j) {
        freq[j]=0;
        for(i=0;i<N;i++) {
            //frequncy of RN within bin width
            if((M[i]>=j*h)&&(M[i]<(j+1)*h)) {
                freq[j]++;
            }
        }
    }
    // stroing frequncy distribution
    FILE*fp=NULL;
    fp=fopen("3c2a.txt","w");
    for(j=0;j<bin;++j) {
        fprintf(fp,"%lf\t%d\n",j*h,freq[j]);
    }
    //Correlation Checks
    FILE*fp1=NULL;
    fp1=fopen("3c2b.txt","w");
    for(i=0;i<N-2;++i) {
        fprintf(fp1,"%lf\t%lf\t%lf\n",M[i],M[i+1],M[i+2]);
    }
}
```



Sun Mar 21 12:08:34 2021



Sun Mar 21 12:09:52 2021

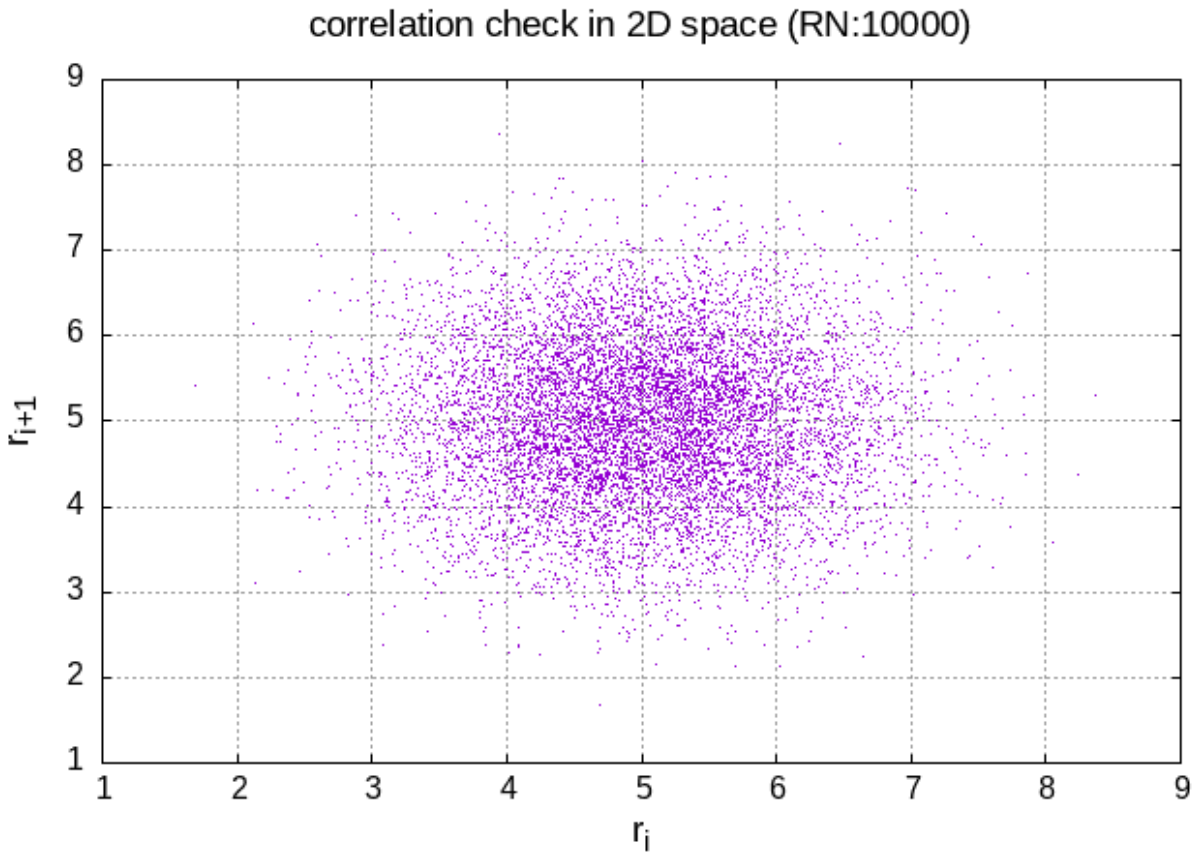


PROBLEM 3 : Case III

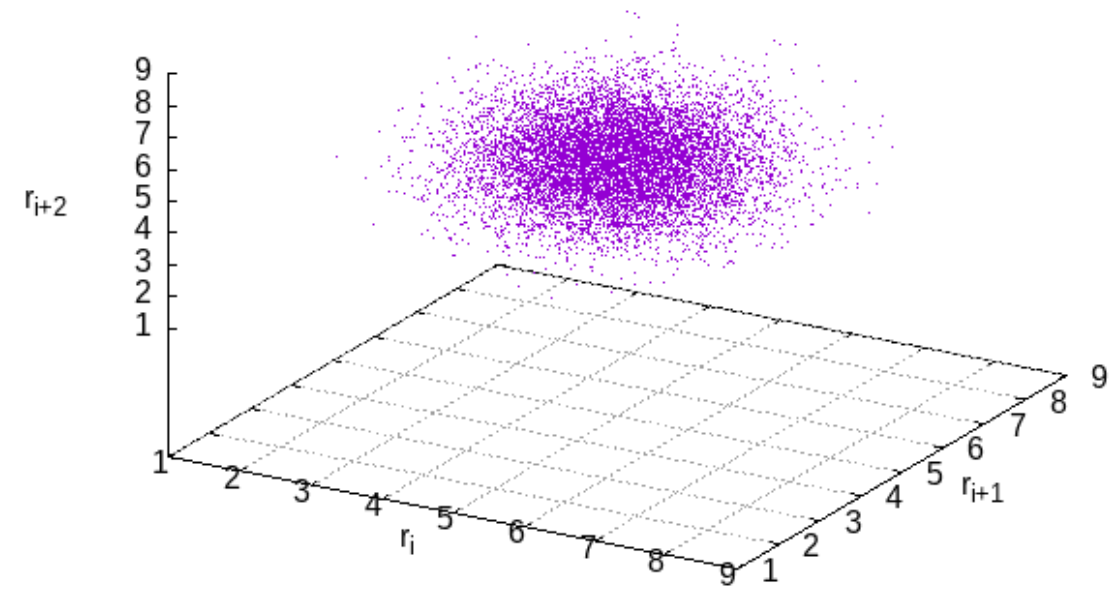
In [ ]:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i,j,N=10000,max=1,sub=10;//subject
    double ai[N],bi[N],ci[N],di[N],ei[N],fi[N],gi[N],hi[N],ji[N],ki[N],M[N];
    //srand(time(0));
    for(i=0;i<N;++i) {
        ai[i]=((double)rand()/RAND_MAX)*max;
        bi[i]=((double)rand()/RAND_MAX)*max;
        ci[i]=((double)rand()/RAND_MAX)*max;
        di[i]=((double)rand()/RAND_MAX)*max;
        ei[i]=((double)rand()/RAND_MAX)*max;
        fi[i]=((double)rand()/RAND_MAX)*max;
        gi[i]=((double)rand()/RAND_MAX)*max;
        hi[i]=((double)rand()/RAND_MAX)*max;
        ji[i]=((double)rand()/RAND_MAX)*max;
        ki[i]=((double)rand()/RAND_MAX)*max;
        M[i]=ai[i]+bi[i]+ci[i]+di[i]+ei[i]+fi[i]+gi[i]+hi[i]+ji[i]+ki[i];
    }
    //frequncy distribution within bin width
    double h=0.1; //width of interval
    int bin=10*sub*max; //10 intervals of width 0.1 in [0:1]
    int freq[bin];
    for(j=0;j<bin;++j) {
        freq[j]=0;
```

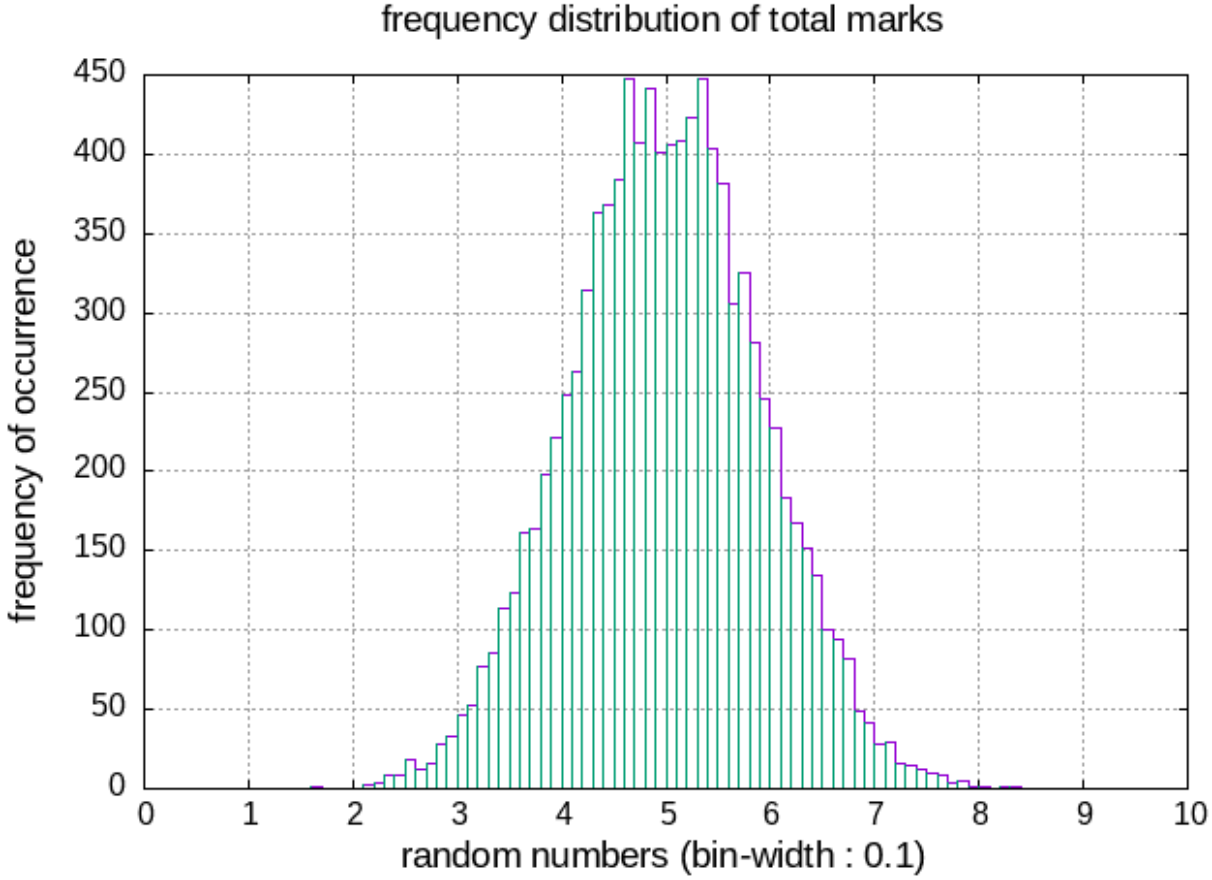
```
        for(i=0;i<N;i++) {
            //frequncy of RN within bin width
            if((M[i]>=j*h)&&(M[i]<(j+1)*h)) {
                freq[j]++;
            }
        }
    }
    // stroing frequncy distribution
    FILE*fp=NULL;
    fp=fopen("3c3a.txt","w");
    for(j=0;j<bin;++j) {
        fprintf(fp,"%lf\t%d\n",j*h,freq[j]);
    }
    //Correlation Checks
    FILE*fp1=NULL;
    fp1=fopen("3c3b.txt","w");
    for(i=0;i<N-2;++i) {
        fprintf(fp1,"%lf\t%lf\t%lf\n",M[i],M[i+1],M[i+2]);
    }
}
```



Sun Mar 21 12:07:54 2021



Sun Mar 21 12:10:06 2021



Sun Mar 21 12:15:25 2021