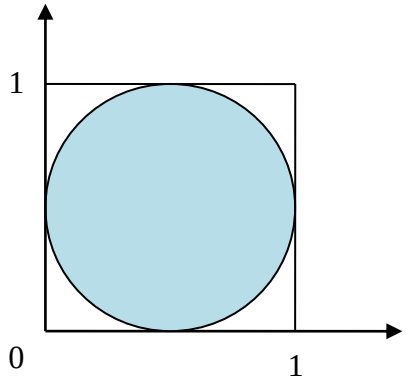
Homework 7

1. Write a program to simulate 6000 rolls of six-sided die and output the frequency of each number.

2. Charlie tosses a pair of six-sided dice. **What number** (sum of the face value of both dice) is **most likely to thrown?** (a 2 is a combination of 1 and 1; a 7 is a combination of 4and 3, 5 and 2, or 6 and 1, and so forth)Please write a program to simulate the process of the toss.

3. Write a program to simulate throwing darts. (射鏢遊戲)



Use a random number generator to obtain 1,000 pairs of floating-point numbers (x, y) satisfying $0< x<1$, $0< y<1$.

Print the proportion P of throws that hit the dart board, that is, the proportion of pairs (x, y) that are inside the circle. Also print $4* P$.

Notice that the geometry of the problem leads us to expect P to be about $\frac{\pi}{4}$. Thus $4* P$ provides an approximation of π.

Note: You can use the following process to generate random number between 0~1:

```
double seed;
const double    mpy = 25173.0;
const double    inc =13849.0;
const double    mod =65535.0;
```

input variable "seed" then calculate the following formula:

seed = (seed *mpy + inc) % mod ; // fmode(seed *mpy + inc, mod)

then **get one random number** between 0~1 by using: **seed/mod**

4. Newton's method (加分題,可不寫.但建議寫且弄清楚)

From Wikipedia, the free encyclopedia You may see more on Wikipedia
This article is about Newton's method for finding roots. For Newton's method for finding minima, see Newton's method in optimization.

In numerical analysis, **Newton's method** (also known as the **Newton – Raphson method**), named after Isaac Newton and Joseph Raphson, is a method for finding successively better approximations to the roots (or zeroes) of a real-valued function.

$$x : f(x) = 0 .$$

The Newton – Raphson method in one variable is implemented as follows:

Given a function f defined over the reals x, and its derivative f', we begin with a first guess x_0 for a root of the function f. Provided the function satisfies all the assumptions made in the derivation of the formula, a better approximation x_1 is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} .$$

Geometrically, $(x_1, 0)$ is the intersection with the x-axis of a line tangent to f at $(x_0, f(x_0))$.

The process is repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

until a sufficiently accurate value is reached.

This algorithm is first in the class of Householder's methods, succeeded by Halley's method. The method can also be extended to complex functions and to systems of equations.

Problem:
Please enter one positive integer from the keyboard and find out the square root of this positive integer.(Using Newton's method only)

5. (加分題,可不寫.但建議寫且弄清楚)

A dog is lost in a tunnel at node 0 (see diagram). It can move one node at one time in either direction right or left with equal probability (1 = right, 2 = left). When the dog hits nodes L_2 however, a force of nature always

propels(推進)him directly to node L_4. The dog escapes from the tunnel when he either hits L_5 or R_4.

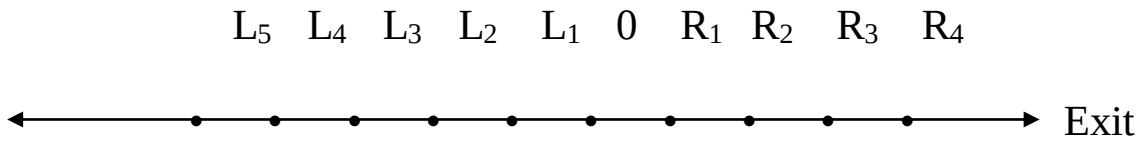
Write a program to determine:

a. Whether the dog has a better chance to exit from the right or the left: In facts, what are the odds (勝算,可能性) that he will exit from R_4? From L_5?

b. How long, on the average, the dog stays in the tunnel (each node takes one minute to cover).

c. Do the same problem as in part a, but let node L_2 propel(推進)the dog to L_4 only when traveling in a left direction. If node L_2 is reached when traveling to the right, the node L_2 has no effect.

Restart the dog at node 0 **a thousand times** and count the number of times he escapes through R_4 or L_5.



6. (加分題,可不寫)

Write a program that displays the name of a card **randomly chosen** from a complete deck of 52 playing cards. Each card consists of a rank (ace, 2,3,4,5,6,7,8,9,10,jack, queen, king) and suit (clubs, diamond, hearts, spades). Your program should display the complete name of the

card, as shown in the following sample run: