Three doors

In an old TV show, three closed doors were presented to a player. A prize was hidden behind one of these doors; there was nothing behind the other two. There is one **player** and a **host**.

The "game" consisted of these steps (one following another):

- The **player** picks one door.
- The **host** does not open this door, but opens another door (there never is anything behind it).
- The **host** offers the **player** the chance of switching the selected door for a fee of $500.
- The **player** rejects the offer and opens the door originally selected **OR** takes the offer, pays $500 and opens the third (remaining) door.
- The **player** wins if the prize is behind the door she opened.

Analysis

There are two possible strategies:

Keep In this strategy, the **player** stays with his/her original pick.

Change In this strategy, the **player** pays $500 and switches to the third door.

Our goal is to decide which of the two strategies is better **on average**. Here “better” means: gives a higher probability of opening the door with a prize.

It is easy to analyse this problem. Suppose we don’t have the patience for such analysis; then we can find out which strategy is better by simulation.

A trivial simulator

At first, we will try to model faithfully the flow of the game:

1. The “winning” door is picked at random (variable **car**).
2. The action of the player is simulated by selecting a door at random (variable **pick**). (Why at random?)
3. An empty door to be opened is picked at random (variable **show**). The possibility that **show = pick** must be avoided.
4. We determine which strategy wins in this scenario and add 1 to the appropriate counter.

```
#include<iostream.h>
extern "C" { long random() ; }

int main()
{
   int Keep = 0  , Change = 0 ;
   int car , pick , show ;
   for( int i = 0 ; i < N ; i++ ) {
      car = random() % 3 + 1 ;
      pick = random() % 3 + 1 ;
      do {
         show = random() % 3 + 1 ;
      } while( show == car || show == pick ) ;
      if( pick == car )
         Keep++ ;
      else
         Change++ ;
   }
   cout << Keep/N << ' ' << Change/N << '\n' ;
}
```

The frequency of strategy #1 being correct:

trials	frequency
10	0.888889
20	0.631579
50	0.408163
60	0.389831
70	0.391304
80	0.405063
100	0.393939

The frequency of strategy #1 being correct:

trials	frequency
100	0.393939
500	0.328657
1000	0.327327
2000	0.330165
5000	0.336667
9000	0.327703
10000	0.330033

Improvement

It is easy to see that at the moment when the variables **car** and **pick** are given values, the outcome is final: if **car = pick**, the player should use strategy #1; otherwise, strategy #2.

With that in mind we note that it is superfluous to have the variable show—it does not change the probability that strategy #1 is correct.

```
#include<iostream.h>
extern "C" { long random() ; }

int main()
{
   int Keep = 0 ;
   int car , pick ;
   for( int i = 0 ; i < N ; i++ ) {
      car = random() % 3 + 1 ;
      pick = random() % 3 + 1 ;
      if( pick == car )
         Keep++ ;
   }
   cout << Keep/N << ' ' << 1 - Keep/N << '\n' ;
}
```

Beware of nasty compiler tricks:

```
#include<iostream.h>
extern "C" { long random() ; }

int main()
{
   int Keep = 0 ;
   for( int i = 0 ; i < N ; i++ )
      if( random() % 3  == random() % 3 )
         Keep++ ;
   cout << Keep/N << ' ' << 1 – Keep/N << '\n' ;
}
```

This will most likely return the output **1 0** because the two cals to **random()** will be ‘optimised” into one call.

Variation

Consider now a different game, similar to the 3–door game:

- The two sides are, as before, the **player** and the **host**.
- There are three cards lying face down on the table. Each card has a different integer written on it (since they are face down, it is impossible for the **player** to tell what these numbers are).
- The **player** picks one card and turns it around.
- The **host** takes the other two cards and puts on the table the lower of the two face up (for everyone to see).
- The **host** offers the **player** the chance of switching cards for a fee of $500.
- The **player** rejects the offer **OR** swaps cards with the **host** and pays $500.
- The **player** wins $2000 if she has a higher card than the **host**.