

## 1. Monty Hall Problem:

(1) (30%) 寫程式前的準備：請先思考如何撰寫你的程式，並把你的程式架構與想法寫下來，愈詳細愈好

首先，寫三個整數變數  $a, b, c$ ，分別隨機給這三個變數 1,2,3，1 為汽車，其餘為羊，設定兩個策略，第一個策略是不要交換，再寫一個迴圈， $i$  最大到 3，若用出來的數字和  $a$  相同，則回傳 1，否則回傳 0，之後把全部的回傳值加總起來，第二個策略是要交換，一樣寫一個迴圈， $i$  最大到 3，若出來的數字和  $a$  相同則回傳 0，否則回傳 1，之後一樣把全部的回傳值加總起來

參考網址:

(如何設定變數 <https://fresh2refresh.com/c-programming/c-variables/>)

(如何設定隨機 <https://blog.gtwang.org/programming/c-cpp-rand-random-number-generation-tutorial-examples/>)

(如何設定隨機 <https://blog.gtwang.org/programming/c-cpp-rand-random-number-generation-tutorial-examples/>)

(如何設定隨機 <http://a7419.pixnet.net/blog/post/96716839-c-rand%28%29-%E4%BA%82%E6%95%B8%E4%BD%BF%E7%94%A8%E6%96%B9%E5%BC%8F>)

寫迴圈可以重複 10000 次

(如何寫迴圈 <https://www.programiz.com/c-programming/c-for-loop>)

兩個策略，

(如何隨機選擇字串 <https://stackoverflow.com/questions/26566186/pick-randomly-string-from-array-in-c>)

Random Number from Array C

[closed]<https://stackoverflow.com/questions/46675878/random-number-from-array-c>

完成版 Code :

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

int main(){
    srand( time(NULL) );
    int i;
    double sum_unchange=0;
    double sum_change=0;
    for (i = 0; i < 10000; i++){
        int a,b,c;//設置變數

        a = 1+rand()%3,b = 1+rand()%3,c = 1+rand()%3;
        while (a==b || b==c || a==c){
            a = 1+rand()%3,b = 1+rand()%3,c = 1+rand()%3;
        }
        int array_1[] = {a,b,c};
        int index = rand()%3;
        int choose = array_1[index];
        double choose_unchange=0;
        double choose_change=0;
        double unchange_array[10000]={};
        double change_array[10000]={};

        if (choose ==1){
            choose_unchange=1.0;
            choose_change=0.0;
        }
        else{
            choose_unchange=0.0;
            choose_change=1.0;
        }
        double temp1=choose_unchange;
        double temp2=choose_change;
        unchange_array[i]=temp1;
        change_array[i]=temp2;
        sum_unchange = sum_unchange+unchange_array[i];
        sum_change = sum_change+change_array[i];
    }
}
```

```
        printf("sum_unchange:%f and sum_change:%f\n",sum_unchange,sum_change);  
  
    }
```

## 2. 100 Prisoners problem

(1) (30%) 寫程式前的準備：請先思考如何撰寫你的程式，並把你的程式架構與想法寫下來，愈詳細愈好

首先產生 1~100 號的向量，然後隨機產生 100 個不重複的抽屜號碼，號碼是 1~100 之間，再寫一個 for loop (i 只有 0 到 10000)重複實驗，另外寫一個迴圈 (給他重複抽 50 次)，迴圈內用一個暫存值等於犯人的號碼然後拿它去抽抽屜，除此之外，迴圈內還有寫 while，while 內容是若抽到的第 i 個抽的和自己號碼一樣則把變數 win(原本是 0)加 1，如果沒有抽到一樣的則跳出 while，暫存一個值然後抽那個暫存值的抽屜號碼，除了暫存值以外的變數加 1，最後加總 win 的數量，最後在 while 內寫 if else，如果 win 的數量是 100 則輸出 winwin，且讓一個空 array 的第 i 個(10000 次重複實驗)變成 1，否則變成 0。

參考網址：

(產生隨機向量 <https://stackoverflow.com/questions/22186423/array-of-random-numbers-using-c-program>)

再寫一個 0~100 的迴圈給犯人抽抽屜

(不要產生重複的值 <https://www.quora.com/How-do-I-generate-random-numbers-without-repetition-in-C>)

[https://www.csie.ntu.edu.tw/~b98902112/cpp\\_and\\_algo/cpp/while\\_loop.html](https://www.csie.ntu.edu.tw/~b98902112/cpp_and_algo/cpp/while_loop.html))

完成版 Code:

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

int main(){
    int i;
    int j;
    int k;
    int use=0;
    int sum_100_win=0;
    float sum_percent=0.0;
    int drawer_array[100]={};
    int criminal_array[100]={};
    int temp_win;
    int temp_win_array[100]={0};
    int sum_100_array[10]={0};
    int sum_win=0;

    srand(time(NULL));
    for (i =0;i<10000;i++){

        for (j = 0; j < 100; j++)
            { int okay = 0;
              criminal_array[j] = j; //創犯人號碼
              temp_win_array[j]=1;
              while (!okay) {
                  drawer_array[j] = (rand() % 100) ;
                  okay = 1;

                  for (k = 0; k < j; k++) {
                      if (drawer_array[k] == drawer_array[j])
                          { okay = 0;
                          }
                      }
                  }
            }
    }
```

```

}

for (int w = 0; w < 100; w++)
{
    use=0;
    int temp_array[100]={};
    temp_array[w]=drawer_array[w];

    while(criminal_array[w]!=temp_array[w]){ u

        se++;
        int temp=0;
        temp =temp_array[w];
        temp_array[w]=drawer_array[temp];

        if(use<50){
            continue;
        }
        if (use==50){
            temp_win_array[w]=0;

            break;
        }
    }
    w=w;
    temp_win=temp_win_array[w];
    sum_100_win=sum_100_win+temp_win;
}
if
    (sum_100_win==100){ s
    um_100_array[i]=1;
}else{
    sum_100_array[i]=0;
}
sum_win=sum_win+sum_100_array[i];
sum_100_win=0;
}
sum_percent=sum_win/10000.0;
printf(" and %f \n",sum_percent);
return 0;

```

}