C 語 言 練 習 - 最 少 派 車 數

某遊覽車派遣公司共收到n筆任務訂單，訂單中詳細記載發車時間s和返回時間d。每一輛遊覽車只要任務時間不衝突，可立即更換司機繼續上路執行任務。請問該公司至少需要調遣多少車輛才足以應付需求？

程式的輸入包含兩行數字，第一行包含一個正整數n，1 ≤ n ≤ 30，代表第二行有n筆訂單的出發時間和返回時間s1, d1, s2, d2, ..., sn, dn，0 < si < di ≤ 24，and si < sj if i<j, 而此2n個正整數間以空格隔開。

問題：請您設計一個程式，輸出最少車輛需求數

我的想法：若前車回來的時間(d) 在下一輛車發車之前(s) (不重疊的情況下)
count+1(同一輛再派的次數) 用全部的訂單扣掉同一輛車派的次數，即為最少派車數，但這樣無法用已用過的車輛。
所以改成，只要有重疊的部分，count+1(改以"重疊"為多一輛車的基準)並用陣列依時間儲存該時間對應的派車數，再用一變數儲存"已派車數"

實際程式碼：

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

int main(int argc, char *argv[])
{
    int n;
    scanf("%d",&n);
    int s[30] = {0};
    int d[30] = {0};
    int time[25] = {0};
    int i = 0,j = 0;
    for(i = 0; i < n; i++)
        scanf("%d %d",&s[i],&d[i]);
    for(i = 0; i < n; i++){
        for(j = s[i]; j < d[i]; j++)
            time[j]++;
    }
    int max = time[0];
    for(i = 0; i < 24; i++){
        if(time[i] > max)
            max = time[i];
    }
    printf("%d\n",max);
    return 0;
}
```

過程所碰到的問題與學習：

這題雖然相較容易，但要將概念實際轉化為程式碼時要注意太多細節，像是陣列宣告的大小。還有最後要用變數儲存"已派車數"我也想了好一下子。這題比較著重在陣列變數之間，釐清完時間跟派車數的關係後，撰寫的過程多注意細節即可。

C 語 言 練 習

WHO IS THE LUCKY GUY

有一個原始部落他們有一個習俗, 就是在執行一群死囚 死刑時會留下一個幸運者給予免死. 作法如下列方式

1. 首先向讓死囚(假設有N個人)排成一列, 編號從1到N (N<100)
2. 進行祭天儀式, 然後巫師向天祈求一個數字 K. 然後將K個小球放入一個球筒中.
3. 從第一位死囚開始每個人依序從球筒內拿出一個球, 拿到最後一顆球的人, 就就地執法.
4. 接下來, 將K個球收回球筒內, 再從下一個人依序從球筒內拿出一個球, 一樣拿到最後一顆球的人, 就就地執法.
5. 這樣的程序一直進行到剩下最後一個人, 那個人就是最後的幸運者, 可以免死離開.

問題：請您設計一個程式, 預估最後的幸運者是誰.

我的想法：直覺的第一個想到是利用N、K間餘數關係

若K>N，用K%N求第一個死。後將死者變為0(生者為1))

再用for loop求下一個一直到最後一個(剩最後一個1)

期間若找到0，則N+1往後一個人。

也可用for人(N)+1；球數(K)-1去找每次球變成0對應的囚犯

實際程式碼：

```
int findNextNonZeroElement(const int a[], const int n, const int offset)
{// linear search from index offset, and find the index k, such that a[k]!=0, return k if
    for(int i=offset;i<n;i++){
        if(a[i]!=0){
            return i;
        }
    }
    for(int i=0;i<offset;i++){
        if(a[i]!=0){
            return i;
        }
    }
    return -1;
}

int findNextVictim_v1(const int a[], const int n, const int nruns, const int offset){
    int k=offset;
    for(int i=0; i<nruns; i++){
        k=findNextNonZeroElement(a, n, k);
        if(k<0)return k;
        if(i<nruns-1) k=(k+1)%n; //if not the last run , start from next; otherwise it is
    }
    return k;
}

int main(void){
    int n; // number of players
    int luckyNumber; // the lucky number;
    int alive[100]; // alive[i]=1/0 means player i is alive/dead
    scanf("%d%d",&n, &luckyNumber);
    n=n%100; //n is less than 100
    assert(n>0);
    for(int i=0; i<n; i++) alive[i]=1; // initially, all player is alive

    for(int i=0, nextStart=0; i<n-1; i++){ // we need to mask of n-1 elements
        //find next victim
        int k=findNextVictim_v1(alive, n, luckyNumber, nextStart);
        assert(k>=0);
        alive[k]=0; // mask it off
        nextStart=(k+1)%n; //star from next one
```

過程所碰到的問題與學習：

在一次for跑完後要接下去重頭抽，必須將排一排抽球的景象轉為圍成圓圈，一直轉到最後一人為止。在函式應用上尚未熟悉，不太了解使用return的時機。nextStart指出下一個開始抽球的人，也就是找下一個非0的人，容易被忽略。

對函式有更深的了解與熟悉外，也認識了const(唯讀)的用法，但對於隱性轉型的部分需更熟練。

PYTHON小練習

剪刀石頭布

猜拳遊戲：玩家按照規則輸入可與電腦猜拳，請寫出一個可以跟電腦猜拳的程式

我的想法：同c語言用簡單的一層while，配合if-elif的應用即可解決

實際程式碼：

```
import random
print("----石頭剪刀布開始----")
print("請按下面規則出拳")
print("石頭[1],剪刀[2],布[3],退出[4]")
while 1:
    user = int(input("請出拳："))
    computer=random.randint(1,3)
    if user==4:
        break
    if ((computer == 1 and user == 3) or (computer == 2 and user
    == 1) or (computer == 3 and user == 2)):
        print("你出{}，電腦出{}，你贏了!!".format(user, computer))
    elif computer == user:
        print("你出{}，電腦出{}，平手!再來啊!".format(user, computer)
        )
    else:
        print("你出{}，電腦出{}，你輸了!菜雞~".format(user, computer)
        )
print("遊戲結束")
```

過程所碰到的問題與學習：

非常簡單的python小練習，對我來說，卻有跟c語言截然不同的感受。python有大量函式庫可以引入，甚至可以crawler，拿網上大量資料去運算，非常方便，語法也精煉了許多。但我認為c語言還是必須先學、有了基礎，學python才能把精力著重在認識更多函式庫的應用上。

C語言期末專題

A SIMPLE DATABASE OPERATIONS

本問題模擬一個小型資料庫操作, 假設有一個資料庫記錄N (0<=N<100) 筆資料, 每一筆資料包含3個欄位: 員工編號 (char employeeId[20], format xxx, from 001 to 999), 姓名 (char name[20]), 薪資 (int salary).
起始時資料庫沒有資料(N=0). 資料庫操作指令格式如下: [指令] [資料], 指令(char cmd)={'a' | 'd' | 'u' | 'q'}, 分別代表加入(add)/刪除(delete)/更新(update)一筆資料; 以及離開(quit)結束操作.
請根據輸入指令操作直到結束(指令='q') 後, 列出目前薪資最高的一筆資料.
指令如下
1. a 員工編號 姓名 薪資 2. d 員工編號 3. u 員工編號 姓名 薪資 4. q

我的想法：以switch(轉接口)代替if-else，增加程式可讀性
quit判斷是0還是1就好
若ID已有人用則return -1
若初始化相同return i，若不同則return -1

實際程式碼：

```
void processingDBcommands(record_t db[])
{
    //possible commands:
    // 1. add a new record: a employeeID name salary
    // 1. delete a new record: d employeeID
    // 1. update a new record: u employeeID name salary
    // 4. quit: q

    char cmdBuff[100];
    record_t rec;
    char cmdStr[20];
    int quit=0;
    while(1){

        //read a line from input
        gets(cmdBuff);
        // parsing the input line
        sscanf(cmdBuff, "%s%s%s%d",cmdStr, rec.employeeId, rec.name, &rec.salary);
        //printf("%s",cmdStr[0]);
        switch(cmdStr[0]){
        case 'a': addRecord(db, rec);  break;
        case 'A': addRecord(db, rec);  break;
        case 'd': deleteRecord(db, rec.employeeId);  break;
        case 'D': deleteRecord(db, rec.employeeId);  break;
        case 'u': updateRecord(db, rec);  break;
        case 'U': updateRecord(db, rec);  break;
        case 'q': quit=1; break;
        case 'Q': quit=1; break;
        //default: // do nothing skip

        }
        if(quit) break;
    }
}

int findMaxSalaryRecord(const record_t db[])
{

    int idx_maxSalary=-1;
```

過程所碰到的問題與學習：

第一次寫那麼複雜的程式，剛開始嘗試卻不知該從何下手，一直碰壁、不懂很多函式的用意。需要不斷找人問，不斷翻書、爬文，去搞懂每一個用法的細節。但這些都是過渡期，撐過去後，熟練就會非常有成就感。這次我有多嘗試使用return值來協助函式的判斷，比上次更熟練。