

10947 Delete linked list

3個function、記得update count

1. NewNode: 記得allocate一個Node的空間
2. CreateList: 四個宣告 *node *new *temp *pre, new記得allocate一個Node的空間, while loop 跑每個input
3. DeleteNode: (*nd)的update要記清楚, for loop跑要delete的target

12602 OuQ String

1. 記得要先用long long array記錄每個相對應 S_k 的總長度, 再用遞迴跑解答
2. 先看 k 是否為1, 是則印OuQ的部分或全部; 否則

```
(0) if ( $index_l \leq 0 \ \&\& \ index_r \geq 0$ ) print("O")
(1) update  $index_l$  and  $index_r$  length
(2) if ( $index_l \leq len[k - 1] - 1 \ \&\& \ index_r \geq 0$ ) recursively call self
(3) update  $index_l$  and  $index_r$  length
(4) if ( $index_l \leq 0 \ \&\& \ index_r \geq 0$ ) print("u")
(5) update  $index_l$  and  $index_r$  length
(6) if ( $index_l \leq len[k - 1] - 1 \ \&\& \ index_r \geq 0$ ) recursively call self
(7) update  $index_l$  and  $index_r$  length
(8) if ( $index_l \leq 0 \ \&\& \ index_r \geq 0$ ) print("Q")
```

12603 Launch a Collider

For loop 看 position[i] 與 position[i + 1]是否反向(R, L), 是的話看看中間距離是否是最短的

12604 N queens and M Rooks Problem

For loop 掃從column = 0 ~ size - 1, 同時要記錄放了幾個queen跟rook, loop中要確認兩個事情:

queen的檢查: $n < N \ \&\& \$ 用check_queen function來確認目前放的位子是否合格(斜線,橫,直)
rook的檢查: $m < M \ \&\& \$ 用check_rook function來確認目前放的位子是否合格(橫,直)

確認完後記得要放旗子上去:

```
cnt += put_chess(r + 1, n + 1, m, N, M, size); (queen)
cnt += put_chess(r + 1, n, m + 1, N, M, size); (rook)
```

12605 Rerendering

scanf(" %c %c", &char1, &char2); 記得%c前要加空格, 這題妳自己寫的應該多看一兩遍就好唷~

12606 Happy New Year

紀錄最遠最近然後差值乘二

12611 The Same Calendar

1. 寫個function確認是否為閏年: `return (!(n % 400) || (!(n % 4) && n % 100)) ? 1 : 0 ;`
2. 用while loop去跑是否與第一年的年性質相同(call 剛剛1裡面的function)以及用same去確認是第一天為同一個星期

```
while(same || is_leap_ori ^ is_leap(year)) {    // is_leap_ori 是否為第一年為閏年
    same += is_leap(++year) ? 366 % 7 : 365 % 7;
    same %= 7;
}
```

12612 Queries on a String

用strncpy去複製substring速度比用for loop快, 用餘數去決定每個character的最後位置

12613 Yet Another Meme Problem

```
printf("%.0f\n", a * floor(log10(b+1)));
```

12614 Game shopping

For loop 跑, 如果current bill的數值大於current game, 總count++

12615 Knight Search

確認以下事情, 則recursively call自己直到目標字串來到最後一個則return成功:

1. $i^2 + j^2$ 是否為五(馬自行走法)
2. 目前的格子是否為目標字母
3. X, Y是否有越界(小於0 or 大於size)