

## 題目 554

### 554. 骰子

[Submit](#)[Status](#)[Discussion / Solution](#)[Ranklist](#)[Back to Problems List](#)

TopCoder



yyy

User's AC Ratio

80.0% (8/10)

Submission's AC Ratio

42.9% (9/21)

Tags

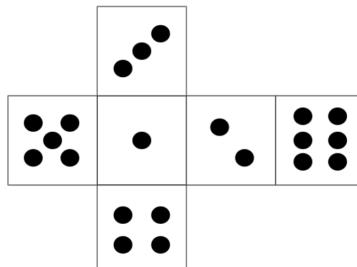
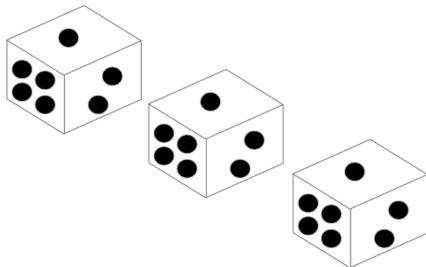
[Show solution-related tags](#)

apsc apsc-array

#### Description

註：本題與 636 相同。

給定  $n$  個骰子排成一列，一開始都是點數 1 朝上，點數 4 朝前，點數 2 朝右（如下圖所示），另外骰子的展開圖如下圖所示。



接下來有  $m$  次修改操作，每個操作包含兩個整數  $a, b$

若  $a, b$  都是正整數，交換編號  $a$  與編號  $b$  的骰子的位置。（ $a$  可能等於  $b$ ）

若  $b$  為  $-1$ ，將編號  $a$  的骰子向前旋轉。（點數 1 朝上 → 點數 3 朝上）

若  $b$  為  $-2$ ，將編號  $a$  的骰子向右旋轉。（點數 1 朝上 → 點數 5 朝上）

在  $m$  次操作結束之後，依序輸出編號 1 到編號  $n$  的骰子朝上的點數。

#### Input Format

第一行包含兩個正整數  $n, m$ 。

接下來  $m$  行每行有兩個整數，第  $i$  行的兩個整數表示第  $i$  次操作。

- $1 \leq n \leq 20$
- $1 \leq m \leq 100$

#### Sample Input 1

```
1 2
1 -2
1 -1
```

#### Output Format

在一行輸出  $n$  個數字以空格分隔，第  $i$  個數字表示編號  $i$  的骰子最後朝上的點數。

#### Sample Input 2

```
3 3
2 -1
3 -2
3 1
```

#### Sample Output 1

```
3
```

#### Hints

#### Sample Output 2

```
5 3 1
```

#### Problem Source

Zerojudge f580

Results of submission #67721						
Back to Submissions List		Total Time (ms)	Max Memory (RSS, KIB) ⓘ	Verdict ⓘ	Score	
554. 骰子		104	3456	Accepted	100	
Subtask Results						
Subtask no.	Testdata Range	Constraints			Score	
1	0~1	範例測資			0 / 0	
2	0~25	無額外限制			100 / 100	
Testdata Results						
Testdata no.	Subtasks	Time (ms)	Memory (VSS, KIB) ⓘ	Memory (RSS, KIB) ⓘ	Verdict ⓘ	Score
0	① ②	3.2	6356	3456	Accepted	100
1	① ②	4.1	6356	3456	Accepted	100
2	②	4.1	6356	3456	Accepted	100
3	②	4.1	6356	3456	Accepted	100
4	②	4.0	6356	3456	Accepted	100
5	②	4.1	6356	3456	Accepted	100
6	②	4.0	6356	3456	Accepted	100
7	②	4.0	6356	3456	Accepted	100
8	②	3.9	6356	3456	Accepted	100
9	②	4.1	6356	3456	Accepted	100
10	②	4.0	6356	3456	Accepted	100
11	②	4.1	6356	3456	Accepted	100
12	②	3.9	6356	3456	Accepted	100
13	②	4.1	6356	3456	Accepted	100
14	②	4.0	6356	3456	Accepted	100
15	②	4.0	6356	3456	Accepted	100
16	②	4.1	6356	3456	Accepted	100
17	②	3.9	6356	3456	Accepted	100

這題的關鍵是任一面與其對面的點數相加為 7，所以只要知道三個面就可以知道骰子的狀態

接下來以 [左 上 右] 代表骰子，假設一顆 [4 1 2] 的骰子做 -1 的動作(向前轉，也就是頂部跑到左邊)，會變成 [上, 左的對面, 右]=[上, 7-左, 右]=[1 3 2]

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    int n, m;
    cin >> n >> m;
    int a = 4;
    int b = 1;
    int c = 2;
    int **arr = new int *[n];
    for (int i = 0; i < n; i++)
    {
        arr[i] = new int [3];
    }
    for(int i=0;i<n;i++){
        arr[i][0] = a;
        arr[i][1] = b;
        arr[i][2] = c;
    }
}
```

```
for (int j = 0; j < m; j++)
{
    int x, y;
    cin >> x >> y;
    x--;
    if (y == -1)
    {
        int temp = arr[x][0];
        arr[x][0] = arr[x][1];
        arr[x][1] = 7 - temp;
    }else if (y == -2)
    {
        int temp = arr[x][2];
        arr[x][2] = arr[x][1];
        arr[x][1] = 7 - temp;
    }else{
        y--;
        int *temp = new int[3];
        for (int k = 0; k < 3; k++)
        {
            temp[k] = arr[x][k];
        }
        for (int k = 0; k < 3; k++)
        {
            arr[x][k] = arr[y][k];
        }
        for (int k = 0; k < 3; k++)
        {
            arr[y][k] = temp[k];
        }
    }
}

for (int i = 0; i < n; i++)
```

```
{  
    cout << arr[i][1] << " ";  
}  
for (int i = 0; i < n; i++)  
    delete[] arr[i];  
delete[] arr;  
  
return 0;  
}
```