

ZeroJudge o712. 2. 蒐集寶石

<https://zerojudge.tw/ShowProblem?problemid=o712>

APCS 113/10

Pei-yih Ting

範例解析

$$M=4$$

$$N=5$$

$$K=2$$

$$r=2$$

$$c=1$$

範例解析

$M=4$

$N=5$

$K=2$

$r=2$

$c=1$

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	3	2	3	0
3	1	1	-1	3	1

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	3	2	3	0
3	1	1	-1	3	1

score

0

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	3	2	3	0
3	1	1	-1	3	1

score

0	3
---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	2	3	0
3	1	1	-1	3	1

score

0	3
---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	2	3	0
3	1	1	-1	3	1

score

0	3	5
---	---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	3	0
3	1	1	-1	3	1

score

0	3	5
---	---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	3	0
3	1	1	-1	3	1

score

0	3	5	8
---	---	---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	2	0
3	1	1	-1	3	1

score

0	3	5	8
---	---	---	---

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	2	0
3	1	1	-1	3	1

score

0	3	5	8	11
---	---	---	---	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	2	0
3	1	1	-1	2	1

score

0	3	5	8	11
---	---	---	---	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	2	0
3	1	1	-1	2	1

score

0	3	5	8	11	13
---	---	---	---	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13
---	---	---	---	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	2	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15
---	---	---	---	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15
---	---	---	---	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	1	1
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15	16
---	---	---	---	----	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	0	1
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15	16
---	---	---	---	----	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	0	1
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15	16	17
---	---	---	---	----	----	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	0	0
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15	16	17
---	---	---	---	----	----	----	----	----

範例解析

M=4

N=5

K=2

r=2

c=1

	0	1	2	3	4
0	2	0	1	0	0
1	2	-1	0	1	-1
2	0	2	1	1	0
3	1	1	-1	2	1

score

0	3	5	8	11	13	15	16	17	17
---	---	---	---	----	----	----	----	----	----

行走規則

行走規則

1. 每到一格**分數**先加上該格**寶石數量**

行走規則

1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就**拿走**一顆寶石，**寶石數量**減 1

行走規則

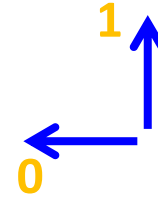
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**

行走規則



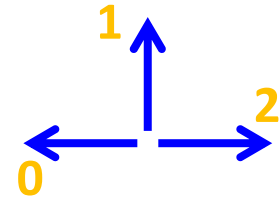
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**

行走規則



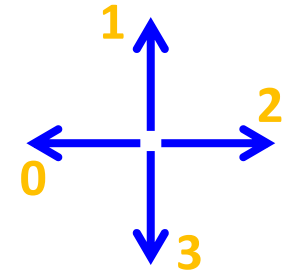
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**

行走規則



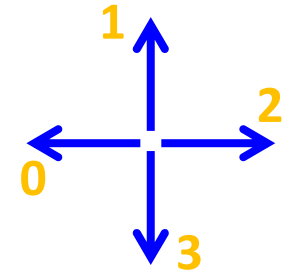
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**

行走規則



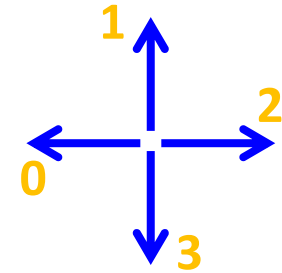
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**

行走規則



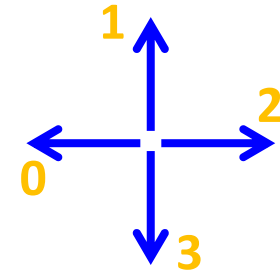
1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**

行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1

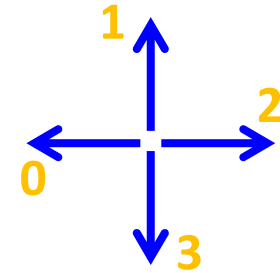
行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1
6. 依照目前方向**前進一格**


```
int dirs[4][2] =
```

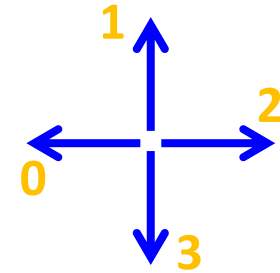
行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就**拿走**一顆**寶石**，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. **嘗試**依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1
6. 依照目前方向**前進一格**

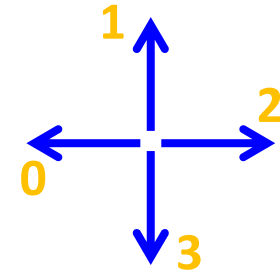
```
int dirs[4][2] = {{0,-1},
```

行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就**拿走**一顆**寶石**，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 **-1** 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 **-1**
6. 依照目前方向**前進一格**

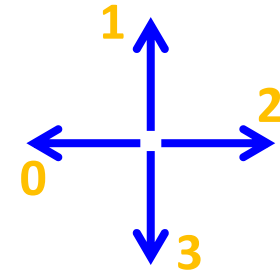
`int dirs[4][2] = {{0,-1},
{-1, 0},` 行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就**拿走**一顆**寶石**，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. **嘗試**依照目前方向前進，如果是 **-1** 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 **-1**
6. 依照目前方向**前進一格**

```
int dirs[4][2] = {{0,-1},  
                 {-1, 0},  
                 {0, 1},
```

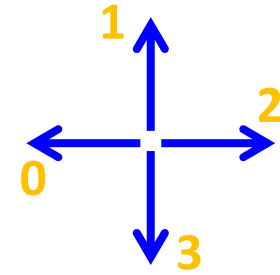
行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就**拿走一顆寶石**，**寶石數量減 1**
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1
6. 依照目前方向**前進一格**

```
int dirs[4][2] = {{0,-1},  
                 {-1, 0},  
                 {0, 1},  
                 {1, 0}};
```

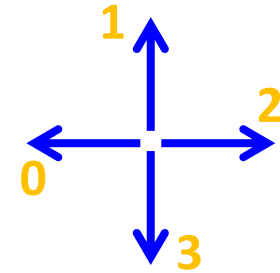
行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1
6. 依照目前方向**前進一格**

```
int dirs[4][2] = {{0,-1},  
                 {-1, 0},  
                 {0, 1},  
                 {1, 0}};
```

行走規則



1. 每到一格**分數**先加上該格**寶石數量**
2. 每到一格就拿走一顆寶石，**寶石數量**減 1
3. 如果分數是 k 的倍數，**轉 90°**
4. 嘗試依照目前方向前進，如果是 -1 或是出界，**轉 90°**
5. 重複步驟 4，直到沒有出界且不為 -1
6. 依照目前方向**前進一格**
7. 重複執行 1-6 所有步驟，直到目前那一格**沒有寶石**