

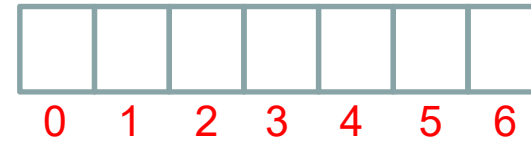
簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

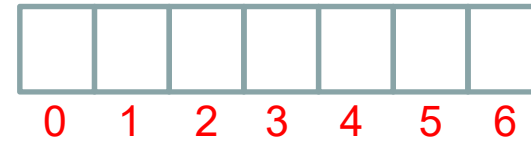
```
01 int n, m, i, j, r, q, p[200000];
```



簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```



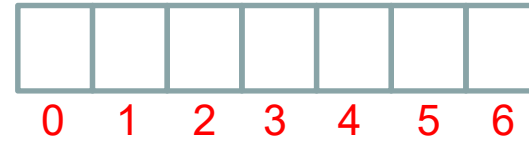
範例一輸入
7 3

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```



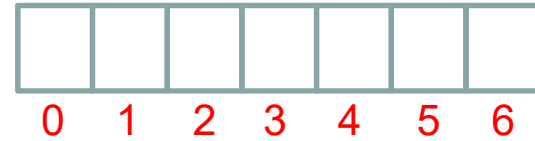
範例一輸入
7 3

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```



範例一輸入
7 3
2 1 5 4 3 5 3

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];  
02 scanf("%d%d", &n, &m);  
03 for (i=0; i<n; i++)  
04     scanf("%d", &p[i]);
```

2	1	5	4	3	5	3
0	1	2	3	4	5	6

範例一輸入
7 3
2 1 5 4 3 5 3

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

```
01 int n, m, i, j, r, q, p[200000];
```

```
02 scanf("%d%d", &n, &m);
```

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

```
04     scanf("%d", &p[i]);
```

2	1	5	4	3	5	3
0	1	2	3	4	5	6

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=n-1, j=0; j<m; j++) {
06     scanf("%d", &q);

09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=n-1, j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; r+=p[i])
08         i=(i+1)%n;
09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=n-1, j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; r+=p[i])
08         i=(i+1)%n;
09 }
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數
- 到達第 n 間房間的時候藉由除以 n 的餘數折返陣列最前端

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=n-1, j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; r+=p[i])
08         i=(i+1)%n;
09 }
10 printf("%d\n", (i+1)%n);
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

範例一輸出

4

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數
- 到達第 n 間房間的時候藉由除以 n 的餘數折返陣列最前端

簡單迴圈基準實作 ... 2

- 以陣列存放 $p_0, p_1, \dots, p_{n-1}$

2	1	5	4	3	5	3
0	1	2	3	4	5	6

```
01 int n, m, i, j, r, q, p[200000];
02 scanf("%d%d", &n, &m);
03 for (i=0; i<n; i++)
04     scanf("%d", &p[i]);
05 for (i=n-1, j=0; j<m; j++) {
06     scanf("%d", &q);
07     for (r=0; r<q; r+=p[i])
08         i=(i+1)%n;
09 }
10 printf("%d\n", (i+1)%n);
```

範例一輸入

7 3

2 1 5 4 3 5 3

8 9 12

範例一輸出

4

- 以迴圈模擬一間一間房間前進
 r 是蒐集到的點數
 - 到達第 n 間房間的時候藉由除以 n 的餘數折返陣列最前端
- 內層迴圈最多把陣列 p 走過一次， n 個元素，外層迴圈執行 m 次， m 個 q 值，所以時間需要 4×10^9 ，會超過 2 秒限制