

| | |
|-----------|-----------------------------------|
| Status | Finished |
| Started | Tuesday, 3 December 2024, 8:28 PM |
| Completed | Friday, 6 December 2024, 12:12 AM |
| Duration | 2 days 3 hours |
| Marks | 60.00/60.00 |
| Grade | 10.00 out of 10.00 (100%) |

Question 1

Correct

Mark 10.00 out of 10.00

[ChangeValue]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Nhập vào 3 số nguyên i, j, x , nhiệm vụ của bạn là thay đổi giá trị của phần tử ở hàng thứ i và cột thứ j của ma trận thành x .

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên n, m ($m, n \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.
- Dòng tiếp theo chứa 3 số nguyên i, j, x , các số liên tiếp nhau cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình m dòng, mỗi dòng gồm n số (cách nhau bởi một dấu cách) biểu diễn ma trận sau khi đã thay đổi giá trị.

For example:

| Input | Result |
|-------|--------|
| 3 3 | 2 3 5 |
| 1 3 5 | 2 4 7 |
| 2 4 7 | 3 4 2 |
| 3 4 2 | |
| 1 1 2 | |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n, m; cin >> m >> n;
17     int a[m][n];
18     for(int i = 0; i < m; i++)
19     {
20         for(int j = 0; j < n; j++)
21         {
22             cin >> a[i][j];
23         }
24     }
25     int x, y, z; cin >> x >> y >> z;
26     a[x-1][y-1] = z;
27     for(int i = 0; i < m; i++)
28     {
29         for(int j = 0; j < n; j++)
30         {
31             cout << a[i][j] << " ";
32         }
33     }

```

```
32     cout << el;
33     }
34     return 0;
35 }
36
37
38
```

| | Input | Expected | Got | |
|---|--|---|--|---|
| ✓ | 3 3 1 3 5 2 4 7 3 4 2 1 1 2 | 2 3 5 2 4 7 3 4 2 | 2 3 5 2 4 7 3 4 2 | ✓ |
| ✓ | 10 20 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 1 5 4 | 610 132 5 743 4 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 | 610 132 5 743 4 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 2

Correct

Mark 10.00 out of 10.00

[ChangePosition]

Cho ma trận kích thước $m \times n$ chứa các số nguyên, các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n .

Viết chương trình nhập vào 4 số nguyên i_1, j_1, i_2, j_2 và trao đổi giá trị của 2 phần tử ở vị trí (i_1, j_1) và (i_2, j_2)

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng.

- Dòng đầu tiên chứa 2 số nguyên n, m ($m, n \leq 1000$);
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, cách nhau bởi một dấu cách;
- Dòng tiếp theo chứa 4 số nguyên i_1, j_1, i_2, j_2 .

Đầu ra

In ra màn hình m dòng, in ra ma trận sau khi đã [đổi chỗ](#).

For example:

| Input | Result |
|---------|--------|
| 3 3 | 4 3 5 |
| 1 3 5 | 2 1 7 |
| 2 4 7 | 3 4 2 |
| 3 4 2 | |
| 1 1 2 2 | |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n, m; cin >> m >> n;
17     int a[m][n];
18     for(int i = 0; i < m; i++)
19     {
20         for(int j = 0; j < n; j++)
21         {
22             cin >> a[i][j];
23         }
24     }
25     int x, y, z, t; cin >> x >> y >> z >> t;
26     swap(a[x-1][y-1], a[z-1][t-1]);
27     for(int i = 0; i < m; i++)
28     {
29         for(int j = 0; j < n; j++)
30         {
31             cout << a[i][j] << " ";
32         }
33     }

```

```
33     cout << endl;
34 }
35 return 0;
36 }
37
38
```

| | Input | Expected | Got | |
|---|---|---|--|---|
| ✓ | 3 3 1 3 5 2 4 7 3 4 2 1 1 2 2 | 4 3 5 2 1 7 3 4 2 | 4 3 5 2 1 7 3 4 2 | ✓ |
| ✓ | 10 20 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 71 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 142 4 3 10 20 | 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 142 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 71 | 610 132 5 743 825 241 790 839 814 26 453 879 348 535 839 668 756 340 604 292 273 609 515 122 96 613 398 440 522 340 592 133 472 597 228 298 190 18 489 5 396 294 236 96 829 75 764 938 767 721 582 40 682 98 163 778 711 561 571 586 901 515 142 725 112 299 375 655 669 216 12 417 510 600 513 692 27 630 630 794 351 564 834 33 662 349 164 726 262 735 312 163 250 383 241 714 34 616 369 703 185 381 120 695 333 985 387 360 615 369 154 966 934 341 352 948 690 516 26 953 251 690 468 501 425 61 215 459 678 937 514 863 318 986 910 652 972 650 364 939 371 871 906 305 212 610 606 254 126 632 207 377 323 28 230 748 89 445 560 119 382 74 334 53 413 597 57 385 247 421 324 618 644 582 276 856 192 882 111 670 866 670 47 189 50 277 290 492 75 850 611 809 276 298 214 689 895 271 426 71 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 3

Correct

Mark 10.00 out of 10.00

[SortInColumn]

Cho ma trận A, kích thước $m \times n$, các hàng được đánh số từ 0 đến $m-1$, các cột được đánh số từ 0 đến $n-1$. Nhập vào một số nguyên x ($x \geq 0$). Hãy sắp xếp lại dữ liệu trên cột x theo thứ tự giảm dần rồi in kết quả ra màn hình. Nếu x không nằm trong đoạn $[0, n-1]$ thì bạn không phải làm gì.

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên n, m ($m, n \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.
- Dòng cuối chứa số nguyên x .

Đầu ra

In ra màn hình ma trận sau khi đã sắp xếp giảm dần cột x .

For example:

| Input | Result |
|-------|--------|
| 3 3 | 1 2 6 |
| 1 2 3 | 4 5 4 |
| 4 5 6 | 2 3 3 |
| 2 3 4 | |
| 2 | |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13
14 int main()
15 {
16     Faster;
17     int m, n; cin >> m >> n;
18     int a[m][n];
19     for(int i = 0; i < m; i++)
20     {
21         for(int j = 0; j < n; j++)
22         {
23             cin >> a[i][j];
24         }
25     }
26     int x; cin >> x;
27     vector<int> vt;
28     for(int i = 0; i < m; i++)
29     {
30         vt.push_back(a[i][x]);

```

```
31     }
32     sort(vt.be, vt.en, greater<int>());
33     int t = 0;
34     for(int i = 0; i < m; i++)
35     {
36         for(int j = 0; j < n; j++)
37         {
38             if(j == x) cout << vt[t++];
39             else cout << a[i][j];
40             cout << " ";
41         }
42         cout << el;
43     }
44     return 0;
45 }
46
```

| | Input | Expected | Got | |
|---|--|--|--|---|
| ✓ | 3 3 1 2 3 4 5 6 2 3 4 2 | 1 2 6 4 5 4 2 3 3 | 1 2 6 4 5 4 2 3 3 | ✓ |
| ✓ | 4 5 1 1 1 1 1 4 1 5 0 6 1 1 1 1 1 1 5 1 1 1 1 | 1 5 1 1 1 4 1 5 0 6 1 1 1 1 1 1 1 1 1 1 | 1 5 1 1 1 4 1 5 0 6 1 1 1 1 1 1 1 1 1 1 | ✓ |
| ✓ | 2 3 2 1 0 3 4 5 0 | 3 1 0 2 4 5 | 3 1 0 2 4 5 | ✓ |
| ✓ | 4 4 3 2 4 2 6 7 8 1 2 3 9 10 10 6 3 100 3 | 3 2 4 100 6 7 8 10 2 3 9 2 10 6 3 1 | 3 2 4 100 6 7 8 10 2 3 9 2 10 6 3 1 | ✓ |
| ✓ | 4 4 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 4 | 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 | 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 4

Correct

Mark 10.00 out of 10.00

[SortInRow]

Cho ma trận A, kích thước $m \times n$, các hàng được đánh số từ 0 đến $m-1$, các cột được đánh số từ 0 đến $n-1$. Nhập vào một số nguyên x ($x \geq 0$). Hãy sắp xếp lại hàng x theo thứ tự tăng dần rồi in kết quả ra màn hình. Nếu x không nằm trong đoạn $[0, m-1]$ thì bạn không phải làm gì.

Đầu vào

Đầu vào từ bàn phím gồm $m + 2$ dòng:

- Dòng đầu tiên chứa 2 số nguyên n, m ($m, n \leq 1000$).
- m dòng tiếp theo mỗi dòng chứa n số nguyên biểu diễn ma trận đã cho, các số liên tiếp nhau trên một dòng cách nhau bởi một dấu cách.
- Dòng cuối chứa số nguyên x .

Đầu ra

In ra màn hình ma trận sau khi đã [sắp xếp tăng dần](#) hàng x .

For example:

| Input | Result |
|-------|--------|
| 3 3 | 1 2 3 |
| 1 2 3 | 4 5 6 |
| 4 5 6 | 2 3 4 |
| 2 3 4 | |
| 2 | |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13
14 int main()
15 {
16     Faster;
17     int m, n; cin >> m >> n;
18     int a[m][n];
19     for(int i = 0; i < m; i++)
20     {
21         for(int j = 0; j < n; j++)
22         {
23             cin >> a[i][j];
24         }
25     }
26     int x; cin >> x;
27     vector<int> vt;
28     for(int i = 0; i < n; i++)
29     {
30         vt.push_back(a[x][i]);

```

```
31     }
32     sort(vt.be, vt.en);
33     int t = 0;
34     for(int i = 0; i < m; i++)
35     {
36         for(int j = 0; j < n; j++)
37         {
38             if(i == x) cout << vt[t++];
39             else cout << a[i][j];
40             cout << " ";
41         }
42         cout << el;
43     }
44     return 0;
45 }
46
```

| | Input | Expected | Got | |
|---|--|--|--|---|
| ✓ | 3 3 1 2 3 4 5 6 2 3 4 2 | 1 2 3 4 5 6 2 3 4 | 1 2 3 4 5 6 2 3 4 | ✓ |
| ✓ | 4 5 1 1 1 1 1 4 1 5 0 6 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 1 0 1 4 5 6 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 1 0 1 4 5 6 1 1 1 1 1 1 1 1 1 1 | ✓ |
| ✓ | 2 3 2 1 0 3 4 5 0 | 0 1 2 3 4 5 | 0 1 2 3 4 5 | ✓ |
| ✓ | 4 4 3 2 4 2 6 7 8 1 2 3 9 10 10 6 3 100 3 | 3 2 4 2 6 7 8 1 2 3 9 10 3 6 10 100 | 3 2 4 2 6 7 8 1 2 3 9 10 3 6 10 100 | ✓ |
| ✓ | 4 4 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 4 | 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 | 3 2 4 2 6 7 8 1 2 3 9 10 3000 6 10 100 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 5

Correct

Mark 10.00 out of 10.00

[AllPossibleWay]

Giả sử bạn là một hành khách đi du lịch đến một đất nước xa lạ.

Tại quốc gia này du lịch rất phát triển, nên chính phủ đã xây dựng một bản đồ điện tử để hỗ trợ khách du lịch.

Trong bản đồ này, cho biết tất cả những tuyến xe khách có thể đi đến thành phố A và xuất phát từ thành phố A đến các thành phố khác bằng cách lưu trữ thông tin trong một [mảng hai chiều](#).

Giá trị tại $(i, j) = 1$ nếu có tuyến xe từ thành phố thứ i đến thành phố thứ j và ngược lại, $(i, j) = 0$

Các tuyến xe này đi một chiều và không có tuyến quay đầu nếu giá trị $(j, i) = 0$.

Viết chương trình hỗ trợ khách du lịch tìm ra:

- Số lượng những tuyến xe khách xuất phát từ thành phố thứ k và
- Số lượng tuyến xe đi từ bất kì thành phố nào trên bản đồ đến thành phố k

Chương trình nhận đầu vào là ma trận vuông cỡ $n \times n$ gồm thông tin tuyến xe của n thành phố và thành phố thứ k cần xem xét ($1 \leq k \leq n$)

For example:

| Input | Result |
|--|--------|
| 4 1 0 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 | 3 3 |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n; cin >> n;
17     int a[n+1][n+1];
18     for(int i = 1; i <= n; i++)
19     {
20         for(int j = 1; j <= n; j++) cin >> a[i][j];
21     }
22     int x, dem1 = 0, dem2 = 0; cin >> x;
23     for(int j = 1; j <= n; j++) if(a[x][j]) dem1++;
24     for(int i = 1; i <= n; i++) if(a[i][x]) dem2++;
25     cout << dem1 << " " << dem2;
26     return 0;
27 }
28

```

| | Input | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | 4 1 0 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 | 3 3 | 3 3 | ✓ |
| ✓ | 10 0 1 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 1 0 0 9 | 6 5 | 6 5 | ✓ |
| ✓ | 15 0 1 0 0 1 1 1 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 1 1 0 1 1 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0 0 1 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 1 1 1 1 9 | 8 8 | 8 8 | ✓ |

| | Input | Expected | Got | |
|---|---|----------|----------|---|
| ✓ | <div>26</div> <div>0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1</div> <div>1 0 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 0</div> <div>1 1 1 1 1 0 0 0 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0</div> <div>0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1</div> <div>1 1 1 1 1 1 0 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 0</div> <div>0 0 0 1 1 0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 0 1</div> <div>0 0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 1 0 0 1 0 0 0 0 0 0</div> <div>1 1 1 0 0 0 0 1 0 1 0 0 1 1 1 0 1 0 1 0 0 1 1 0 1 0</div> <div>0 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 1 0</div> <div>0 0 1 0 0 1 0 1 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1</div> <div>0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 1 0</div> <div>0 0 1 1 1 0 0 0 0 1 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1</div> <div>0 0 0 1 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 0 0</div> <div>1 0 0 1 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 1 0</div> <div>1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0</div> <div>0 0 1 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0 0</div> <div>0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0 0 0</div> <div>0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0</div> <div>0 0 1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0 1 0</div> <div>1 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0</div> <div>1 0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 1 0 1</div> <div>1 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 0 0</div> <div>1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1</div> <div>0 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0</div> <div>0 1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 1</div> <div>1 1 0 1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0</div> <div>19</div> | 14 12 | 14 12 | ✓ |

| | Input | Expected | Got | |
|---|---|----------|----------|---|
| ✓ | 63 0 1 1 0 1 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 0 0 0 1 0 0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 1 0 0 1 0 1 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 1 1 1 1 1 1 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 1 1 0 1 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 | 37 29 | 37 29 | ✓ |

| Input | Expected | Got | |
|---|----------|-----|--|
| 1 0 0 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 1 1 0 0 1 0 1 0 0 1 1 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 1 1 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 1 1 0 1 1 1 0 1 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1 1 1 1 0 1 0 0 0 | | | |

| | Input | Expected | Got |
|--|---|----------|-----|
| | <div>0 1 0 1 1 1 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1</div> <div>1 0 1 0 0 1 0 1 1 0 0 1 1 0 0 0</div> <div>1 1 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0</div> <div>0 0 1 1 0 1 1 1 0 1 1 1 0 0 0 0</div> <div>1 1 0 0 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 0 0 1 1</div> <div>0 0 1 0 1 0 0 1 0 0 1 1 1 0 1 0</div> <div>1 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1 0</div> <div>0 0 0 0 1 1 0 1 1 0 1 0 0 0 1 0</div> <div>0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 0 1 0 1 0 1 1 0 1</div> <div>0 1 0 1 1 0 0 0 1 1 1 1 1 1 1 0</div> <div>0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 1</div> <div>0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 1 1</div> <div>0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 0</div> <div>1 0 1 0 0 1 1 0 1 1 0 0 1 1 0 0</div> <div>0 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1</div> <div>0 0 1 0 0 0 0 1 0 1 1 1 1 0 0 1</div> <div>1 0 1 1 0 1 1 1 1 0 1 0 1 1 0 1 0 1 0 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 1 1 1</div> <div>0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 1</div> <div>0 0 0 0 1 0 1 0 1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0</div> <div>1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0</div> <div>0 0 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0</div> <div>1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1</div> <div>1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0</div> <div>0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 0</div> <div>0 0 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 1 0 1 1 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 0</div> <div>1 0 1 0 1 1 1 0 1 1 1 0 0 1 0 0</div> <div>34</div> | | |

| | Input | Expected | Got | |
|---|---|----------|----------|---|
| ✓ | <div>34</div> <div>0 0 0 1 1 1 0 0 1 1 1 0 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0</div> <div>1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 0 1 1 0</div> <div>0 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 1 1 0</div> <div>0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 0 1 1 1 1 1</div> <div>1 1 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0</div> <div>0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 0 0 0</div> <div>0 1 0 0 0 1 1 1 1 0 0 0 1 0 0 1 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1</div> <div>1 1 0 0 1 0 1 1 1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1</div> <div>1 0 1 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 1</div> <div>0 0 0 0 1 0 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0</div> <div>0 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1 1 1 1 1 1</div> <div>0 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 1 0 1 1 0 0 1</div> <div>1 0 0 1 1 0 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1</div> <div>1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 0 1</div> <div>0 0 0 1 0 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 0</div> <div>1 0 0 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1 0 1 0 0 1 1 0 0 1 1 1 0 1 1 0 0</div> <div>0 1 1 0 0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0</div> <div>0 0 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0</div> <div>0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0</div> <div>0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0 1 0 1 1 1 0 1 0 0</div> <div>1 0 1 0 1 1 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 1 0</div> <div>1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 1</div> <div>1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0</div> <div>0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1</div> <div>1 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1</div> <div>1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1</div> <div>0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 1 0 1 0 1 0 1</div> <div>1 0 0 1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 1 1 0</div> <div>1 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0</div> <div>0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1 1 0 1 1</div> <div>1 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 1</div> <div>0 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 1 0 1 1 1 0 0</div> <div>1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0</div> <div>0 0 1 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 1</div> | 19 16 | 19 16 | ✓ |
| | <div>34</div> | | | |

34

| | Input | Expected | Got | |
|---|---|----------|-------|---|
| ✓ | <div>34</div> <div>1 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 1</div> <div>0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1</div> <div>1 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 0 0</div> <div>1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 0 1 1 0 1 0 1 0 0 1 0 0</div> <div>0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1</div> <div>1 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 1 1 0</div> <div>1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0 1 1 0 1 1 0 1 1</div> <div>1 1 0 1 1 0 0 0 0 1 0 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0 1 0 1 0</div> <div>0 1 1 1 0 0 0 1 1 0 1 0 0 1 1 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 0</div> <div>0 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 1 1 1 1</div> <div>1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1</div> <div>0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0</div> <div>1 1 0 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0</div> <div>0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1</div> <div>0 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1</div> <div>1 0 0 1 1 1 1 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 0 1 0</div> <div>1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1</div> <div>0 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 1 0 0 1 1</div> <div>1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0</div> <div>1 1 0 0 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1</div> <div>1 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0 0</div> <div>1 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0</div> <div>1 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0</div> <div>0 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 0 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 0 1</div> <div>1 0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0</div> <div>0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0 1</div> <div>0 0 1 1 0 1 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 1</div> <div>0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 1 1 1 1 1 0</div> <div>1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 0 1 1 1 1</div> <div>1 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 1 0 0 1 0</div> <div>1 0 1 0 1 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 0 1 0 1 1</div> <div>0 1 1 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1</div> <div>1 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 1 0 0 1 1 0</div> <div>0 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0</div> <div>34</div> | 12 17 | 12 17 | ✓ |

34

| | Input | Expected | Got | |
|---|--|----------|----------|---|
| ✓ | 26 0 0 0 1 1 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 1 0 1 1 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 0 1 0 0 1 1 0 0 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 0 1 1 19 | 14 15 | 14 15 | ✓ |
| ✓ | 15 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 9 | 6 8 | 6 8 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 6

Correct

Mark 10.00 out of 10.00

[EmployeeInformation]

Thông tin của nhân viên trong công ty được lưu trong một cơ sở dữ liệu dưới dạng bảng 2 chiều.

Mỗi hàng tương ứng với một nhân viên trong công ty.

Các cột lần lượt chứa thông tin về: Giới tính (Nam : 1, Nữ: 0), Tuổi, Tình trạng hôn nhân (Kết hôn: 1, Chưa kết hôn: 0).

Dựa vào bảng thông tin của n nhân viên trong công ty được cung cấp, hãy in ra màn hình số lượng nhân viên là Nữ, trên 30 tuổi và Đã kết hôn.

For example:

| Input | Result |
|--------|--------|
| 5 | 1 |
| 0 23 1 | |
| 1 28 1 | |
| 1 29 1 | |
| 0 29 0 | |
| 0 34 1 | |

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n, ans = 0; cin >> n;
17     int a[n+1][4];
18     for(int i = 1; i <= n; i++)
19     {
20         for(int j = 1; j <= 3; j++) cin >> a[i][j];
21     }
22     for(int i = 1; i <= n; i++)
23     {
24         if(a[i][1] == 0 && a[i][2] >= 30 && a[i][3] == 1) ans++;
25     }
26     cout << ans;
27     return 0;
28 }
29

```

| | Input | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | 9 0 23 1 1 28 0 1 29 1 0 19 0 1 34 1 1 23 0 0 35 1 0 28 1 0 32 1 | 2 | 2 | ✓ |
| ✓ | 11 0 23 1 1 28 1 1 29 1 0 29 0 0 34 1 1 23 0 1 35 1 0 28 1 0 32 1 1 19 0 0 34 1 | 3 | 3 | ✓ |
| ✓ | 5 0 23 1 1 28 1 1 29 1 0 29 0 0 34 1 | 1 | 1 | ✓ |
| ✓ | 5 0 23 1 1 28 1 1 29 1 0 29 0 0 24 1 | 0 | 0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)