



Wybory bitmistrza



Wskazówka

Wczytujemy ilość kandydatów k i ilość miast n .

Tworzymy listę $W[]$ zawierającą ilości zwycięstw poszczególnych kandydatów (na początku wypełnioną zerami). Rozmiar listy to $k + 1$, aby występowały w niej elementy o numerach $W[1], W[2], \dots, W[k]$.

Wykonujemy pętlę po miastach (są to kolejne wiersze danych wejściowych). Dla każdego miasta znajdujemy, który kandydat zdobył w nim najwięcej głosów. Należy tutaj pamiętać o dwóch szczegółach:

- W danym mieście mogą zostać nieoddane żadne głosy, wtedy wynik 0 jest uważany za najlepszy. Dlatego początkowa wartość zmiennej `max_vote` przy wyszukiwaniu musi być ujemna.
- Jeśli przeglądając wyniki kolejnych kandydatów napotkamy kogoś, kto ma tyle samo głosów, ile wynosi wynik uważany przez nas na najlepszy (do tej pory w danym mieście – zmienna `max_vote`), wtedy nie bierzemy go pod uwagę, gdyż w razie remisu liczy się wcześniejszy kandydat (o niższym numerze).

Teraz dla każdego kandydata mamy ilość jego zwycięstw – w liście $W[]$. Przeglądamy ją, wyszukując największy element `max_win` – i jego pozycję `max_i`: to będzie numer kandydata-elekta. Tutaj również liczy się pierwszy kandydat, który uzyska maksymalną ilość głosów. (Wartość początkowa zmiennej `max_win` może być równa 0 lub mniejsza.)

Kod programu w języku C++

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n, k;
    cin >> k >> n;
    vector<int> W(k+1);
    for(int j=1; j<=n; j++)
    {
        int max_vote = -1, max_i;
        for(int i=1; i<=k; i++)
        {
```

```

    int v; cin >> v;
    if(v > max_vote)
    {
        max_vote = v; max_i = i;
    }
}
W[max_i]++;
}
int max_win = 0, max_i;
for(int i=1; i<=k; i++)
    if(W[i] > max_win)
    {
        max_win = W[i]; max_i = i;
    }
cout << max_i << endl;
}

```

Kod programu w języku Python

```

k, n = map(int, input().split())
W = [0] * (k+1);
for j in range(1, n+1):
    V = [0] + [int(v) for v in input().split()]
    max_vote = -1
    max_i = 0 # dowolna wartość
    for i in range(1, k+1):
        if(V[i] > max_vote):
            max_vote = V[i]
            max_i = i
    W[max_i] += 1
max_win = 0
max_i = 0
for i in range(1, k+1):
    if(W[i] > max_win):
        max_win = W[i]
        max_i = i
print(max_i)

```

Uwagi

W całych wyborach może zostać nieoddany ani jeden głos – *notabene* wtedy wygrywa kandydat 1. Tak dzieje się również, gdy wszyscy kandydaci dostaną po tyle samo głosów we wszystkich miastach (dlaczego?).