



# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

## Национален кръг, Хасково, 8-10 март 2024 г.

### Група D, 6 клас

#### Задача D6. Класиране (Пояснение към решението)

1. Наивният подход, при който сортираме всички стойности от входа и след това преброяваме неповтарящите се, ще получи по-малко от половината точки за задачата, защото поради ограничението за паметта от 2 МВ, не може при по-големите тестове да бъдат натоварени всички данни. За да сортираме в намаляващ ред, записваме отрицателни стойности във вектора  $a$ :

```
cin >> n >> k;
for(int i=0;i<n;i++) {int v; cin >> v; a.push_back(-v);}
sort(a.begin(),a.end());
```

За да преброим колко са неповтарящите се стойности във вектора  $a$  след сортирането, използваме следния фрагмент:

```
int c=0, p=0;
for(int i=0;i<a.size();i++)
{
    if(a[i]!=p) {c++;p=a[i];}
    if(c==k) break;
}
```

Накрая отпечатваме: `cout << -p << endl;`.

2. За да решим задачата за 100 т., използваме подход, при който във всеки момент трябва да са натоварени в паметта не повече от 1000 стойности – защото накрая трябва да изведем точките на състезателите, класирани на място  $k$ , където  $k < 1000$ . Използваме масив  $r[]$  с такъв малък размер и чрез функцията `insert(int v)` за всяка следваща прочетена стойност от входа поддържа масива в сортиран вид, като в масива остават само най-много първите  $k$  от прочетените до момента стойности:

```
int c,n,k;
void insert(int v)
{
    if(v<r[c] && c+1 < k) {c++; r[c]=v; return;}
    int i=c;
    while(i>=0 && v>r[i] ) i--;
    if(v==r[i]) return;
    for(int j=c+1;j>i;j--) r[j]=r[j-1];
    r[i+1]=v;
    if(c+1 < k) c++;
}
```

Главната част на програмата е:

```
cin >> n >> k;
int v; cin >> v; r[0]=v;
for(int i=1;i<n;i++) {cin >> v; insert(v);}
cout << r[c] << endl;
```

*Емил Келеведжиев*