Data Science Meeting

30 Mar 2023, 14:13:04
Mar 30, 2023, 10:34:26
start: AM
Mar 30, 2023, 03:34:26
finish: PM
to finish: 01:21:15
...

Jury announcements | Complete

Tasks Parcels Messages

C. Recommendation feed

Time limit	2 seconds
Memory limit	64Mb
Input	standard input or input.txt
Conclusion	standard output or output.txt

Developer Ilya created a social network with a feed of recommendations for heterogeneous content. Various objects (photos, news, videos, etc.) are mixed in the feed. Ilya set up the feed in such a way that objects are ordered by relevance. The more likes an object gets, the closer it is to the beginning of the search. Ilya did not foresee one important condition, that with such an ordering, several objects of the same type in a row often occur in the list of recommendations.
Users don't like it because the recommendations look the same. It is necessary to implement an algorithm that, according to the list of recommendations, will make a new list - devoid of this drawback and at the same time as relevant as possible.

Let the initial list of recommendations be given$a = [a_0, a_1, \ldots, a_{n-1}]$ long$n > 0$. Object numbered$i$ has type with number$b_i \in \{0, \ldots, m-1\}$ and relevance$r(a_i) = 2^{-i}$.

Consider a list that is obtained from the original by choosing a subset of objects and rearranging them: $x = [a_{i_0}, a_{i_1}, \ldots, a_{i_{k-1}}]$ length$k$ ($1 \le k \le n$). A list is called admissible if no two consecutive objects in it have the same type, i.e.$b_{i_j} \ne b_{i_{j+1}}$ for all$j = 0, \ldots, k-2$. The relevance of the list is calculated by the formula$\sum_{j=0}^{k-1} 2^{-j} r(a_{i_j})$. You need to find the most relevant list among all valid ones.

Input Format

The first line contains numbers separated by spaces.n And$m$ ($1 \le n \le 100\,000$, $1 \le m \le n$).
In the next$n$ lines contain numbersb_i For$i = 0, \ldots, n-1$($0 \le b_i \le m-1$).

Output Format

Write down the numbers of the objects of the final list separated by a space:$i_0, i_1, \ldots, i_{k-1}$.

Example 1

Input	Conclusion
eleven 0	0

Example 2

Input	Conclusion
2 2 1 1	0

Example 3

Input	Conclusion

```
10 2                                0 3 1 4 2 6 5
1
1
1
0
0
1
0
1
1
1
```

Language GNU GCC 12.2 C++20

Dial here | Send file

```
1
```

Send | 100 tries left

Previous | Next

No parcels