

```
bfs(competitive_programming)
```

Introduction to Competitive Programming

Get started with competitive programming - part 2

Balajiganapathi S

code-drills.com

April 23, 2017

Outline

- 1 Structure of a problem statement
- 2 Problem types
- 3 Common mistakes
- 4 Conclusion

Outline

- 1 **Structure of a problem statement**
- 2 Problem types
- 3 Common mistakes
- 4 Conclusion

Statement

http://codeforces.com/problemset/problem/796/A

The screenshot shows the Codeforces website interface. At the top, there's a navigation bar with 'HOME', 'CONTESTS', 'GYM', 'PROBLEMSET', 'GROUPS', 'RATING', 'API', 'RCC', 'VK CUP', and 'HFT BATTLE'. The 'PROBLEMSET' tab is active. Below the navigation bar, there's a search bar and a 'PROBLEMS' button. The main content area displays the problem statement for 'A. Buying A House'. The problem description includes the title, time and memory limits, input/output specifications, and a paragraph about Zane and a village. There's an illustration of a girl and a boy with houses. The right sidebar shows 'Codeforces Round #408 (Div. 2)' as 'Finished', a 'Practice' button, and a 'Virtual participation' section with a 'Start virtual contest' button. At the bottom, there's a 'Submit?' section with a language dropdown set to 'GNU G++14 6.2.0', a 'Choose file' button, and a 'Submit' button. The footer of the page shows navigation icons and a green checkmark.

CODEFORCES
Sponsored by Telegram

HOME CONTESTS GYM **PROBLEMSET** GROUPS RATING API RCC VK CUP HFT BATTLE

PROBLEMS SUBMIT STATUS STANDINGS CUSTOM TEST

A. Buying A House

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Zane the wizard had never loved anyone before, until he fell in love with a girl, whose name remains unknown to us.

The girl lives in house m of a village. There are n houses in that village, lining in a straight line from left to right: house 1, house 2, ..., house n . The village is also well-structured: house i and house $i + 1$ ($1 \leq i < n$) are exactly 10 meters away. In this village, some houses are occupied, and some are not. Indeed, unoccupied houses can be purchased.

You will be given n integers a_1, a_2, \dots, a_n that denote the availability and the prices of the houses. If house i is occupied, and therefore cannot be bought, then a_i equals 0. Otherwise, house i can be bought, and a_i represents the money required to buy it, in dollars.

As Zane has only k dollars to spare, it becomes a challenge for him to choose the house to purchase, so that he could live as near as possible to his crush. Help Zane determine the minimum distance from his crush's house to some house he can afford, to help him succeed in his love.

Input

The first line contains three integers n, m, k ($2 \leq n \leq 100, 1 \leq m \leq n, 1 \leq k \leq 100$) — the number of houses in the village, the house where the girl lives, and the amount of money Zane has (in dollars), respectively.

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 100$) — denoting the availability and the prices of the houses.

It is guaranteed that $a_m = 0$ and that it is possible to purchase some house with no more than k dollars.

Codeforces Round #408 (Div. 2)
Finished
Practice
★

Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you — solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you — solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

Submit?

Language: GNU G++14 6.2.0

Choose file: Choose file No file chosen

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

Contest materials

Statement - Story

A. Buying A House

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Zane the wizard had never loved anyone before, until he fell in love with a girl, whose name remains unknown to us.



The girl lives in house m of a village. There are n houses in that village, lining in a straight line from left to right: house 1, house 2, ..., house n . The village is also well-structured: house i and house $i + 1$ ($1 \leq i < n$) are exactly 10 meters away. In this village, some houses are occupied, and some are not. Indeed, unoccupied houses can be purchased.

You will be given n integers a_1, a_2, \dots, a_n that denote the availability and the prices of the houses. If house i is occupied, and therefore cannot be bought, then a_i equals 0. Otherwise, house i can be bought, and a_i represents the money required to buy it, in dollars.

As Zane has only k dollars to spare, it becomes a challenge for him to choose the house to purchase, so that he could live as near as possible to his crush. Help Zane determine the minimum distance from his crush's house to some house he can afford, to help him succeed in his love.

Statement - Task



The girl lives in house m of a village. There are n houses in that village, lining in a straight line from left to right: house 1, house 2, ..., house n . The village is also well-structured: house i and house $i + 1$ ($1 \leq i < n$) are exactly 10 meters away. In this village, some houses are occupied, and some are not. Indeed, unoccupied houses can be purchased.

You will be given n integers a_1, a_2, \dots, a_n that denote the availability and the prices of the houses. If house i is occupied, and therefore cannot be bought, then a_i equals 0. Otherwise, house i can be bought, and a_i represents the money required to buy it, in dollars.

As Zane has only k dollars to spare, it becomes a challenge for him to choose the house to purchase, so that he could live as near as possible to his crush. Help Zane determine the minimum distance from his crush's house to some house he can afford, to help him succeed in his love.

Statement - Input format

Input

The first line contains three integers n , m , and k ($2 \leq n \leq 100$, $1 \leq m \leq n$, $1 \leq k \leq 100$) — the number of houses in the village, the house where the girl lives, and the amount of money Zane has (in dollars), respectively.

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 100$) — denoting the availability and the prices of the houses.

It is guaranteed that $a_m = 0$ and that it is possible to purchase some house with no more than k dollars.

Statement - Output format

Output

Print one integer — the minimum distance, in meters, from the house where the girl Zane likes lives to the house Zane can buy.

Statement - Examples

Examples

input

```
5 1 20
0 27 32 21 19
```

output

```
40
```

input

```
7 3 50
62 0 0 0 99 33 22
```

output

```
30
```

input

```
10 5 100
1 0 1 0 0 0 0 0 1 1
```

output

```
20
```

Note

In the first sample, with $k = 20$ dollars, Zane can buy only house 5. The distance from house $m = 1$ to house 5 is $10 + 10 + 10 + 10 = 40$ meters.

In the second sample, Zane can buy houses 6 and 7. It is better to buy house 6 than house 7, since house $m = 3$ and house 6 are only 30 meters away, while house $m = 3$ and house 7 are 40 meters away.

Statement - Time limit

A. Buying A House

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Zane the wizard had never loved anyone before, until he fell in love with a girl, whose name remains unknown to us.

Statement - solution

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n, m, k, a, ans = 100000;
6     cin >> n >> m >> k;
7     for(int i = 1; i <= n; i++) {
8         cin >> a;
9         if(a != 0 && a <= k) ans = min(ans, 10 * abs(m - i));
10    }
11    cout << ans << endl;
12
13    return 0;
14 }
```

Outline

- 1 Structure of a problem statement
- 2 Problem types**
 - Problem types - scoring
 - Problem types - output
- 3 Common mistakes
- 4 Conclusion

Binary/classical

- Either correct or wrong no partial
- Most common form

Subtasks

- Multiple subtasks with different constraints
- Each subtasks has some points

Subtasks - example

<https://www.codechef.com/problems/XENTASK>

Constraints

Subtask 1: 40 points

- $1 \leq T \leq 10$
- $1 \leq N \leq 3$
- $1 \leq X_i, Y_i \leq 10^5$

Subtask 2: 60 points

- $1 \leq T \leq 10$
- $1 \leq N \leq 2 \cdot 10^4$
- $1 \leq X_i, Y_i \leq 10^5$

Approximate

- There can be many valid solution
- Each valid output may be assigned a score

Approximate - example

<https://www.codechef.com/problems/FACTORIZ>

Scoring

For each individual file, your score will be calculated as the sum of M^2 over all the test cases in this file. Your score for the problem is the average of individual testcases' files scores. The number of points you'll get in the ranklist will be equal to $\text{YourScore}/\text{BestScore}$, where **YourScore** is naturally your score and **BestScore** is the best score, gained so far in this problem.

Please note that the score you get during the competition is the score on the 20% of the test data. The score on the complete test data will be available after the contest.



Multiple outputs

- Multiple valid outputs possible

Multiple outputs

- Multiple valid outputs possible

<http://codeforces.com/problemset/problem/534/A>

Output

In the first line print integer k — the maximum number of students who can be seated so that no two students with adjacent numbers sit next to each other.

In the second line print k distinct integers a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$), where a_i is the number of the student on the i -th position. The students on adjacent positions mustn't have adjacent numbers. Formally, the following should be true: $|a_i - a_{i+1}| \neq 1$ for all i from 1 to $k - 1$.

If there are several possible answers, output any of them.

Problem types - interactive

- Output and input are mixed

<http://codeforces.com/gym/101021/problem/A>

Input

Use standard input to read the responses to the queries.

The input will contain responses to your queries — strings "<" and ">=". The i -th string is a response to the i -th your query. When your program will guess the number print "! x", where x is the answer and terminate your program.

The testing system will allow you to read the response on the query only after your program print the query for the system and perform `flush` operation.

Output

To make the queries your program must use standard output.

Your program must print the queries — integer numbers x_i ($1 \leq x_i \leq 10^6$), one query per line. After printing each line your program must perform operation `flush`.

Each of the values x_i mean the query to the testing system. The response to the query will be given in the input file after you flush output. In case your program guessed the number X , print string "! x", where X — is the answer, and terminate your program.

Outline

- 1 Structure of a problem statement
- 2 Problem types
- 3 Common mistakes**
- 4 Conclusion

Common mistakes - printing message

```
1 int main() {  
2     int t;  
3     cout << "Enter number of test cases"; // DO NOT DO THIS  
4     cin >> t;  
5     // ...  
6 }
```



Common mistakes - not returning 0

```
1 int main() {  
2     //...  
3  
4     return 0; // DO THIS  
5 }
```

Common mistakes - wrong I/O format

- # of test cases

Common mistakes - wrong I/O format

- # of test cases
- Outputting doubles, newlines etc.

Common mistakes - wrong I/O format

- # of test cases
- Outputting doubles, newlines etc.
- Input whole lines, strings etc.

Common mistakes - edge cases

- Only testing on sample case

Common mistakes - edge cases

- Only testing on sample case
- Pay attention to constraints and input sections

Outline

- 1 Structure of a problem statement
- 2 Problem types
- 3 Common mistakes
- 4 Conclusion**

Thank you