Hong Kong Girls' Olympiad in Informatics 2024/25

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
G251	Diploma Checker	1.000 s	256 MB	8 + 8 + 11 + 14 + 12 + 17 + 24 + 6
G252	Maths or Music?	1.000 s	256 MB	5 + 10 + 21 + 23 + 26 + 15
G253	Navigation	1.000 s	256 MB	4 + 5 + 6 + 9 + 11 + 14 + 17 + 16 + 18
G254	Pyramidal Sequence	1.000 s	256 MB	8 + 7 + 11 + 10 + 17 + 20 + 27
G255	Gerrymandering	1.000 s	256 MB	10 + 15 + 16 + 21 + 24 + 14
G256	Smart Reservoir	1.000 s	256 MB	16 + 13 + 15 + 23 + 21 + 12

Notice:
Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In C++ it is `long long` and its token for `scanf`/`printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

Organiser
香港電腦教育學會 The Hong Kong Association for Computer Education

Co-organiser
CityU Department of Computer Science 香港城市大學 City University of Hong Kong

Sponsor
Jane Street

G251 - DIPLOMA CHECKER

Time Limit: 1.000 s / Memory Limit: 256 MB

The International Baccalaureate Diploma Programme (IBDP) is a two-year educational programme for high school students. All students enrolled in the IBDP must complete the following:

- **6 subjects**, with each subject contributing **between 1 and 7 points** (inclusive) towards their final score, and
- the **core component**, contributing **between 0 and 3 points** (inclusive) to their final score.

As a result, a student's total score in IBDP equals the **sum of points of their 6 subjects**, plus the **number of core component points**. A student can achieve a maximum score of $6 \times 7 + 3 = 45$ points in the IBDP.

A student's **number of core component points** is determined by their Theory of Knowledge (TOK) and Extended Essay (EE) grades, which are both **letters from A to E**. Put simply, given a student's TOK grade X, and their EE grade Y, their number of core component points is given by the following table:

$X \setminus Y$	A	B	C	D	E
A	3	3	2	2	Fail
B	3	2	2	1	Fail
C	2	2	1	0	Fail
D	2	1	0	0	Fail
E	Fail	Fail	Fail	Fail	Fail

For example, if a student got a grade of C in TOK and a grade of A in EE, the student has 2 points in the core component.

In this task, a student is considered to have **failed** the IBDP if **any** of the following conditions apply:

1. The student got a grade of E in **at least one** of TOK or EE.
2. The student scored **exactly 1 point** in **any** of the 6 subjects.
3. The student's total score is **less than 24**. (Note that a score of exactly 24 is considered a pass.)

Note: Here, the rules for failing the IBDP are simplified, so they differ slightly from the official IBDP rules.

Alice is an IBDP candidate who has just received her final results. She received $A_1, A_2, A_3, A_4, A_5, A_6$ points for the 6 subjects she took, while her grades in TOK and EE are X and Y respectively. Can you determine whether Alice passed the IBDP or not, and if she passed, what is her total score?

INPUT

The input consists of 8 lines in total.

The i-th of the first 6 lines $(1 \le i \le 6)$ contains a single integer A_i, denoting Alice's score for the i-th subject.

The next line contains a single character X, denoting Alice's TOK grade.

The next line contains a single character Y, denoting Alice's EE grade.

OUTPUT

If Alice failed the IBDP, output `Fail` on the first and only line.

Otherwise, output `Pass` on the first line. Then, output Alice's total score on the second line.

SAMPLE TESTS

	Input	Output
1	1 1 1 1 1 1 D D	Fail

Alice failed the IBDP since she scored 1 point in at least one of her subjects (in fact, every subject).

	Input	Output
2	2 4 7 3 5 7 D D	Pass 28

With TOK grade $X =$ $\boxed{\texttt{D}}$ and EE grade $Y =$ $\boxed{\texttt{D}}$, Alice has 0 core points. Hence, her total score is $2 + 4 + 7 + 3 + 5 + 7 + 0 = 28$.

	Input	Output
3	7 7 7 7 7 1 D D	Fail

	Input	Output
4	4 4 4 3 3 3 D A	Fail

With TOK grade $X =$ $\boxed{\texttt{D}}$ and EE grade $Y =$ $\boxed{\texttt{A}}$, Alice has 2 core points. Hence, her total score is $4 + 4 + 4 + 3 + 3 + 3 + 2 = 23$, which is smaller than 24. So Alice failed the IBDP.

	Input	Output

5	`4`
	`4`
	`4`
	`3`
	`3`
	`3`
	`B`
	`A`
		`Pass`
		`24`

With TOK grade $X =$ `B` and EE grade $Y =$ `A`, Alice has 3 core points. Hence, her total score is $4+4+4+3+3+3+3 = 24$, just enough to pass.

	Input	Output

6	`7`
	`6`
	`7`
	`7`
	`5`
	`7`
	`D`
	`E`
		`Fail`

SUBTASKS

For all cases:
$1 \leq A_1, A_2, A_3, A_4, A_5, A_6 \leq 7$
X and Y are one of the following characters: `A`, `B`, `C`, `D`, `E`.

	Points	Constraints
1	8	$A_1 = A_2 = A_3 = A_4 = A_5 = A_6 = 1$ $X = Y =$ `D`
2	8	$A_1 = A_2 = A_3 = A_4 = A_5 = A_6 = 7$ $X = Y =$ `D`
3	11	$2 \leq A_1 \leq 7$ $A_2 = A_3 = A_4 = A_5 = A_6 = 7$ $X = Y =$ `D`
4	14	It is guaranteed that the first line of the output is `Pass` $X = Y =$ `D`
5	12	$2 \leq A_1, A_2, A_3, A_4, A_5, A_6 \leq 7$ $X = Y =$ `D`
6	17	$X = Y =$ `D`
7	24	It is guaranteed that the first line of the output is `Pass`
8	6	No additional constraints

G252 - MATHS OR MUSIC?

Time Limit: 1.000 s / Memory Limit: 256 MB

In the web-based game Wordle, players have six chances to figure out a secret 5-letter word. Each guess is followed by color-coded feedback, indicating which letters appear in the secret word and whether they're in the correct spots.

Alice and Bob are playing Wordle in person! In this game, there is a secret 5-letter word W. Bob knows what W is, while Alice does not. She only knows that the secret word, W, is guaranteed to be either `MATHS` or `MUSIC`.

Alice wants to find out which of the two possibilities W is. She tells Bob her *guess*, which is in the form of a string S consisting of 5 **distinct letters** (weirdly enough, Alice's guess S is not limited to the two possible secret words, but rather *any* string with 5 distinct letters). Afterwards, Bob gives her a length-5 feedback string T in a way similar to Wordle. The string T only consists of the letters `G`, `Y` and `B`, corresponding to the colours green, yellow and black respectively. Specifically, for $1 \leq i \leq 5$,

- If the i-th letter of the guess appears in W at the correct spot, then $T_i =$ `G`.
- If the i-th letter of the guess appears in W but **not** at the correct spot, then $T_i =$ `Y`.
- If the i-th letter of the guess does not appear in W at any spot, then $T_i =$ `B`.

For example, consider deducing the feedback string T when $S =$ `ADMIN` and $W =$ `MUSIC`.

- The character $S_1 =$ `A` does not appear in W at any spot. So, $T_1 =$ `B`.
- The character $S_2 =$ `D` does not appear in W at any spot. So, $T_2 =$ `B`.
- The character $S_3 =$ `M` appears in W but not at the correct spot. So, $T_3 =$ `Y`.
- The character $S_4 =$ `I` appears in W at the correct spot. So, $T_4 =$ `G`.
- The character $S_5 =$ `N` does not appear in W at any spot. So, $T_5 =$ `B`.

Therefore, $T =$ `BBYGB` in the above example.

Given Alice's guess S and the feedback string T, can you figure out whether the secret word W is `MATHS` or `MUSIC`? It is possible that the information given is insufficient to conclude whether W is `MATHS` or `MUSIC`, in such case, Alice's guess would be a rather poor one...

INPUT

The first line of the input contains a single string S, Alice's guess.

The second line of the input contains a single string T, the feedback string.

OUTPUT

If the secret word W must be `MATHS`, output `MATHS`.

If the secret word W must be `MUSIC`, output `MUSIC`.

If the information given is insufficient to conclude whether W is `MATHS` or `MUSIC`, output `UNSURE`.

SAMPLE TESTS

	Input	Output
1	ADMIN BBYGB	MUSIC

This sample is explained in the problem statement.

	Input	Output
2	MAGIC GBBGG	MUSIC

The secret word must be `MUSIC`, as the guess `MAGIC` matches its first, fourth and fifth characters.

	Input	Output
3	MTHSA GYYYY	MATHS

This sample satisfies the constraints of subtask 4.

	Input	Output
4	MACRO GBYBB	MUSIC
5	HKGOI YBBBB	MATHS
6	QRZYK BBBBB	UNSURE

SUBTASKS

For all cases:
S consists of exactly 5 **distinct** uppercase letters
T consists of exactly 5 uppercase letters
T only consists of the letters `G`, `Y`, `B`
At least one possible secret word is consistent with T

	Points	Constraints
1	5	$T =$ `GGGGG`
2	10	$S =$ `HANDS`
3	21	There exists at least one index i $(2 \leq i \leq 5)$ such that $T_i =$ `G`
4	23	T only consists of the letters `G` and `Y`
5	26	The letters `M` and `S` never appear in Alice's guess S
6	15	No additional constraints

G253 - NAVIGATION

Time Limit: 1.000 s / Memory Limit: 256 MB

Bob is a courier in Hackerland, where he urgently needs to deliver a package as quickly as possible!

Hackerland can be visualized as an infinite 2D grid, as shown in Figure 1. Initially, Bob is located at (X_1, Y_1). He can make multiple moves to reach his destination at (X_2, Y_2). The destination is not the same as his starting point, i.e. $(X_1, Y_1) \neq (X_2, Y_2)$. For each move, he first picks a direction: north, east, south, or west. Then, he either takes a **small step** and moves 1 unit in that direction or takes a **big step** and moves K units in that direction. K is either 1, 3, or 5.

For example, if Bob is at $(-2, 5)$ and K is 3, he can move to:

- $(-2, 6)$, if he takes a small step north
- $(-2, 8)$, if he takes a big step north
- $(-1, 5)$, if he takes a small step east
- $(1, 5)$, if he takes a big step east
- ...

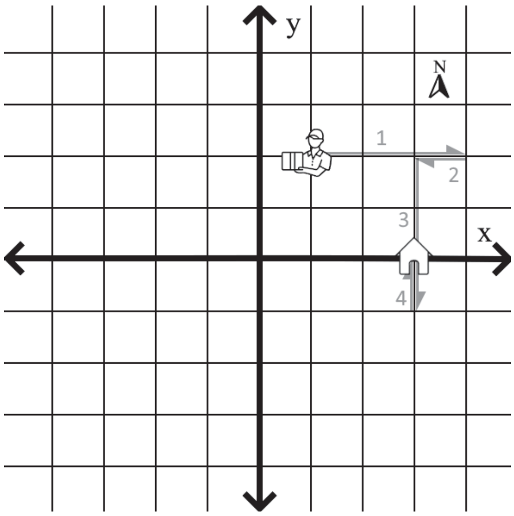


Figure 1: Example configuration of Hackerland. Bob is at $(X_1, Y_1) = (1, 2)$, and the destination is at $(X_2, Y_2) = (3, 0)$. It also shows an example for Bob's move sequence.

To represent the moves, each move is associated with a corresponding character. The letter itself represents the first letter of the direction's name, and we use lowercase to indicate a **small step** and uppercase to indicate a **big step**. Note that if $K = 1$, both a small step and a big step represent the same move. The possible moves are summarized as follows:

Move	Description
`n`	Moves north by 1 unit, i.e., from (x, y) to $(x, y+1)$
`N`	Moves north by K units, i.e., from (x, y) to $(x, y+K)$
`e`	Moves east by 1 unit, i.e., from (x, y) to $(x+1, y)$
`E`	Moves east by K units, i.e., from (x, y) to $(x+K, y)$
`s`	Moves south by 1 unit, i.e., from (x, y) to $(x, y-1)$
`S`	Moves south by K units, i.e., from (x, y) to $(x, y-K)$
`w`	Moves west by 1 unit, i.e., from (x, y) to $(x-1, y)$
`W`	Moves west by K units, i.e., from (x, y) to $(x-K, y)$

Consider the example in Figure 1, if $(X_1, Y_1) = (1, 2)$, $(X_2, Y_2) = (3, 0)$, and $K = 3$, Bob might move according to sequences such as:

- `EwSn`; he moves as follows: $(1, 2) \to (4, 2) \to (3, 2) \to (3, -1) \to (3, 0)$ (Please refer to Figure 1 for the visualisation of the path)
- `eeessw`; he moves as follows: $(1, 2) \to (2, 2) \to (3, 2) \to (4, 2) \to (4, 1) \to (4, 0) \to (3, 0)$
- ...

The first sequence has a length of 4, while the second sequence has a length of 6.

Time is running short. As a friend of Bob, can you help him write a program to find a sequence of moves to go from (X_1, Y_1) to (X_2, Y_2) with the **minimum length**?

INPUT

The first two lines of the input contain two integers X_1 and Y_1, the starting coordinates of Bob.
The following two lines of the input contain two integers X_2 and Y_2, the coordinates of Bob's destination.
The last line of the input contains an integer K.

OUTPUT

Output the sequence of moves without any separation between each move. Every character should be one of `EeSsWwNn`. If there are multiple valid answers, any of them is acceptable.

SAMPLE TESTS

	Input	Output
1	1 2 3 0 3	EwSn

Refer to Figure 1: Bob needs to go from $(1, 2)$ to $(3, 0)$, and his big step size is 3. Other valid sequences with length 4 are also acceptable, for instance `Ewss` or `eess`.

	Input	Output
2	96 0 100 0 5	Ew

	Input	Output
3	2 3 -2 4 5	Wen

	Input	Output
4	0 0 8 0 3	EEEw

This sample meets the constraints of subtask 6.

SUBTASKS

For all cases:
$-100 \leq X_1, Y_1, X_2, Y_2 \leq 100$
$(X_1, Y_1) \neq (X_2, Y_2)$
$K = 1, 3$ or 5

	Points	Constraints
1	4	$K = 1$ $(X_1, Y_1) = (8, 11)$ $(X_2, Y_2) = (8, 3)$
2	5	$K = 3$ $(X_1, Y_1) = (0, 0)$ $(X_2, Y_2) = (5, 0)$
3	6	$K = 5$ $(X_1, Y_1) = (-6, 3)$ $(X_2, Y_2) = (5, 17)$
4	9	$K = 1$ $X_2 > 0$ $X_1 = Y_1 = Y_2 = 0$
5	11	$K = 1$
6	14	$K = 3$ $X_2 > 0$ $X_1 = Y_1 = Y_2 = 0$
7	17	$K = 3$
8	16	$K = 5$ $X_2 > 0$ $X_1 = Y_1 = Y_2 = 0$
9	18	$K = 5$

G254 - PYRAMIDAL SEQUENCE

Time Limit: 1.000 s / Memory Limit: 256 MB

In the depths of an ancient, forgotten forest, you stumble upon a hidden temple, cloaked in moss and tangled vines. The stone walls bear intricate pyramidal patterns — an enigmatic puzzle left by ancient scholars. These patterns reveal a sequence, known as the ***N-pyramidal sequence***, which forms the basis of a challenge.

An N**-pyramidal sequence** begins at 1, increases to N, then decreases back to 1, creating a symmetrical pyramid shape. This sequence repeats infinitely.

For example:

- If $N = 4$, the sequence looks like: 1, 2, 3, 4, 3, 2, 1, 2, 3, 4, 3, 2, 1, ...
- If $N = 2$, the sequence looks like: 1, 2, 1, 2, 1, 2, ...

During your exploration, you uncover a scroll with a list of M numbers, $A = [A_1, A_2, \ldots, A_M]$. You are certain that A appears as a **subsequence** within the infinite N-pyramidal sequence. In this subsequence, each element of A must appear in the same order as in A, though not necessarily consecutively.

For example, consider the case where $N = 4$ and $A = [2, 4, 3, 2, 1]$. One way to locate A as a subsequence in the N-pyramidal sequence is by choosing indices $[2, 4, 5, 6, 7]$ from the sequence $1, \underline{2}, 3, \underline{4}, \underline{3}, \underline{2}, \underline{1}, 2, 3, 4, 3, 2, 1, \ldots$.

As you may have noticed, there are infinitely many ways to locate A. Your goal is to find A as a subsequence in the N-pyramidal sequence such that A_M, the last element of A, appears at the smallest possible index.

INPUT

The first line of the input contains two integers N and M, describing the type of pyramidal sequence and the length of A.
The second line of the input contains M integers $A_1, A_2, \ldots, A_M$, the sequence of numbers you found on the scroll.

OUTPUT

Output a single integer, the minimum index in the N-pyramidal sequence where A_M (the last element of A) can appear.

SAMPLE TESTS

	Input	Output
1	4 5 2 4 3 2 1	7
2	4 3 3 1 2	8

One of the ways to achieve the answer is $1, 2, 3, 4, \underline{3}, 2, \underline{1}, \underline{2}, \ldots$.

SUBTASKS

For all cases:
$2 \le N \le 10^9$
$1 \le M \le 10^5$
$1 \le A_i \le N$

	Points	Constraints
1	8	$N, M \le 100$ $A_1 = A_2 = \cdots = A_M = 1$
2	7	$N, M \le 100$ $A_1 = A_2 = \cdots = A_M = 1$ or $A_1 = A_2 = \cdots = A_M = N$
3	11	$N = 2$
4	10	$N, M \le 100$ $A_1 = A_2 = \cdots = A_M$
5	17	$N = 3$
6	20	$A_i \ne A_{i+1}$ for $1 \le i < M$
7	27	No additional constraints

G255 - GERRYMANDERING

Time Limit: 1.000 s / Memory Limit: 256 MB

In the district of Hackerville, an election is underway with N voters poised to cast their ballots. It is guaranteed that N is a perfect power of 2, i.e. $N = 2^p$ for some positive integer p.

The voting preferences of all voters are represented by an array of N binary digits. Each voter, numbered from 1 to N, will either support Alice (denoted by 1) or oppose her (denoted by 0).

The electoral system in Hackerville operates on a district basis. Initially, there is a single district spanning from voter 1 to voter N, denoted as $(1, N)$. Alice can repeatedly subdivide any district with at least two voters into two equal halves any number of times (possibly none). Specifically, a district defined by the range (l, r) can be subdivided into (l, m) and $(m+1, r)$, where $m = \dfrac{l+r-1}{2}$.

For example, let $N = 8$. Initially, there is only one district $(1, 8)$. Then, $(1, 2), (3, 4), (5, 8)$ is a valid partitioning (achieved by subdividing $(1, 8)$ into $(1, 4), (5, 8)$ and further subdividing $(1, 4)$ into $(1, 2), (3, 4)$), while $(1, 2), (3, 6), (7, 8)$ isn't.

In Hackerville, the outcome of a district is "winner-takes-all"; Alice either wins or loses the whole district. More formally,

- If the number of voters supporting Alice in a district is at least half the district's size ($\left\lceil \dfrac{x}{2} \right\rceil$ supporting votes for a district of x voters), Alice will win the district and gain all x votes.
- Otherwise, Alice will lose that district and won't gain any votes.

For example, consider $N = 4$ voters with voting preferences $[0, 1, 0, 0]$. One possible partitioning for Alice is $(1, 2), (3, 3), (4, 4)$:

- In district $(1, 2)$, Alice receives 1 vote, hence winning the district and getting 2 votes.
- In district $(3, 3)$, Alice receives 0 votes, hence losing the district.
- In district $(4, 4)$, Alice receives 0 votes, hence losing the district.

Hence Alice gets 2 votes in total. It can be shown that there are no valid partitions that achieves more than 2 votes for Alice.

Alice's objective is to maximize the total number of votes she receives by optimally partitioning the districts. Given the array of voters' voting preferences, determine the maximum number of votes Alice can obtain by optimally partitioning the districts. Additionally, provide a valid district partitioning that achieves this maximum.

INPUT

The first line of input consists of an integer N, the number of voters.

The second line contains N integers $A_1, A_2, \ldots, A_N$, the voters' voting preferences.

OUTPUT

On the first line, output an integer V, the maximum number of votes Alice can obtain.

On the second line, output an integer M ($1 \le M \le N$), the number of distinct districts that Alice wants to partition into.

Then output M lines, the i-th line consisting of two integers L_i, R_i, the range of voters for this district (L_i, R_i). Each voter should only belong to exactly one district. You can output the districts in any order.

If there are multiple partitions that satisfy the requirement, output any.

SAMPLE TESTS

	Input	Output
1	4 0 1 0 0	2 3 1 2 3 3 4 4

This sample is explained in the problem statement.

	Input	Output
2	4 0 1 1 0	4 1 1 4

One possible partitioning for Alice is $(1, 4)$, getting 4 votes in total:

- In district $(1, 4)$, Alice receives 2 votes, hence winning the district and getting 4 votes.

	Input	Output
3	16 0 1 1 0 1 0 0 0 0 0 1 0 1 1 1 0	14 5 1 4 5 6 7 7 8 8 9 16

One possible partitioning for Alice is $(1, 4), (5, 6), (7, 8), (9, 16)$, getting 14 votes in total:

- In district $(1, 4)$, Alice receives 2 votes, hence winning the district and getting 4 votes.
- In district $(5, 6)$, Alice receives 1 vote, hence winning the district and getting 2 votes.
- In district $(7, 7)$, Alice receives 0 votes, hence losing the district.
- In district $(8, 8)$, Alice receives 0 votes, hence losing the district.
- In district $(9, 16)$, Alice receives 4 votes, hence winning the district and getting 8 votes.

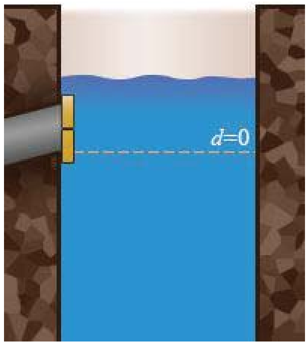
SUBTASKS

For all cases:
$N = 2^p$, where p is a positive integer and $1 \le p \le 17$
$A_i = 0$ or 1 for all $1 \le i \le N$

	Points	Constraints
1	10	$N = 2$
2	15	$N = 4$
3	16	The number of votes for Alice is one of 0 or 1
4	21	The number of votes for Alice is one of 0, 1 or 2
5	24	There exists an index $1 \le k < N$ such that $A[1] = A[2] = \cdots = A[k] = 1$ and $A[k+1] = A[k+2] = \cdots = A[N] = 0$
6	14	No additional constraints

G256 - SMART RESERVOIR

Time Limit: 1.000 s / Memory Limit: 256 MB

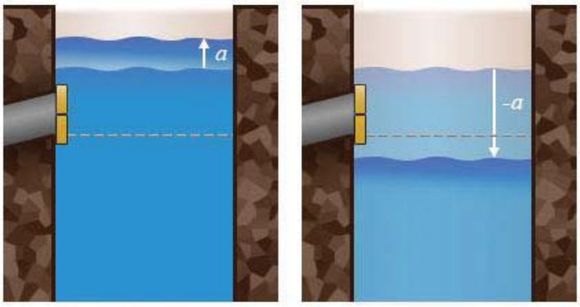


The water levels of a reservoir could change every day. This also applies to Hackerland's newly built reservoir named the Smart Reservoir. The Smart Reservoir is infinitely tall and deep, with a tunnel that channels water out. The reservoir and the tunnel are separated by a gate, which is closed by default. Therefore, water does not flow out from the reservoir normally.

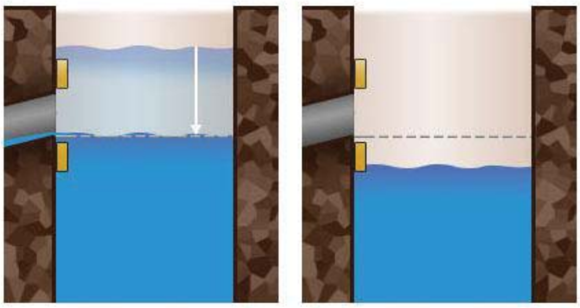
The water level of the reservoir is represented with a value d, its altitude **relative to the gate**. If $d > 0$, it means that the water level is higher than the altitude of the gate. If $d < 0$, it means that the water level is lower than the altitude of the gate.

You already know that there will be N events happening. Each event can be of one of the following types:

- **Change event**: A value a is given, where $a \neq 0$. The water level of the reservoir changes by a value of a. If $a > 0$, the water level rises by a. If $a < 0$, the water level drops by $-a$. Formally, this event changes the water level of the reservoir from d to $d + a$.



- **Gate open event**: This event has a value of 0. The gate will be opened, and water will flow out from the reservoir until the water level reaches the height of the gate. If the water level of the reservoir is already lower than the altitude of the gate, water does not flow out and thus the water level remains unchanged. The gate will be closed automatically before the next event. Formally, this event changes the water level of the reservoir from d to 0 if and only if $d > 0$, and does not change the value of d otherwise.



Everything has been set in stone, except that you do not know the initial water level of the reservoir. However, you are curious to know the final water level of the reservoir after the N events. Therefore, you imagine Q different scenarios. In the i-th scenario, you would like to know the final water level of the reservoir if the initial water level is X_i.

INPUT

The first line of input consists of two integers N and Q, the number of events and the number of scenarios.

The second line contains N integers $A_1, A_2, \ldots, A_N$, where A_i represents the value of the i-th event. The value corresponds to a change event with value A_i if $A_i \neq 0$, and it corresponds to a gate open event if $A_i = 0$.

The next Q lines each contains an integer X_i, the initial water level of the reservoir for the i-th scenario.

OUTPUT

Output Q lines. The i-th line represents the final water level of the reservoir in the i-th scenario.

SAMPLE TESTS

1

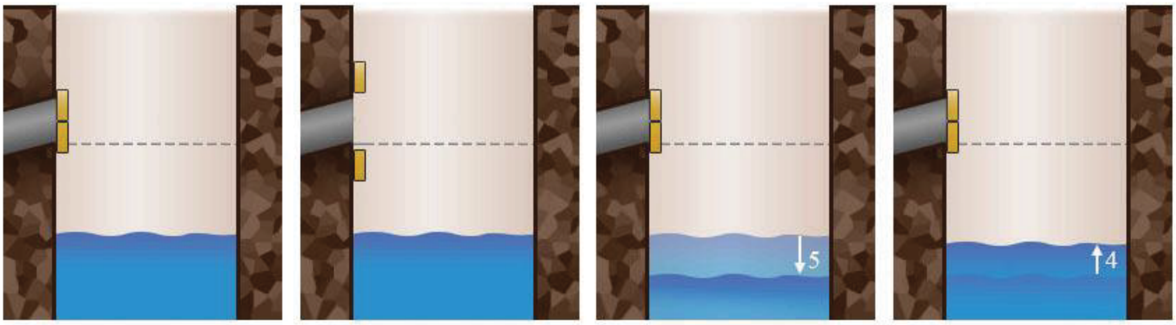
Input	Output

`3 2`
`0 -5 4`
`-10`
`10`
	`-11`
	`-1`

For the first scenario, the initial water level of the reservoir is -10.

- The first event is a gate open event. Since $-10 < 0$, the water level remains to be -10.
- After that, a change event with value -5 happens. The water level becomes $-10 + (-5) = -15$.
- Finally, another change event with value 4 happens. The water level becomes $-15 + 4 = -11$.

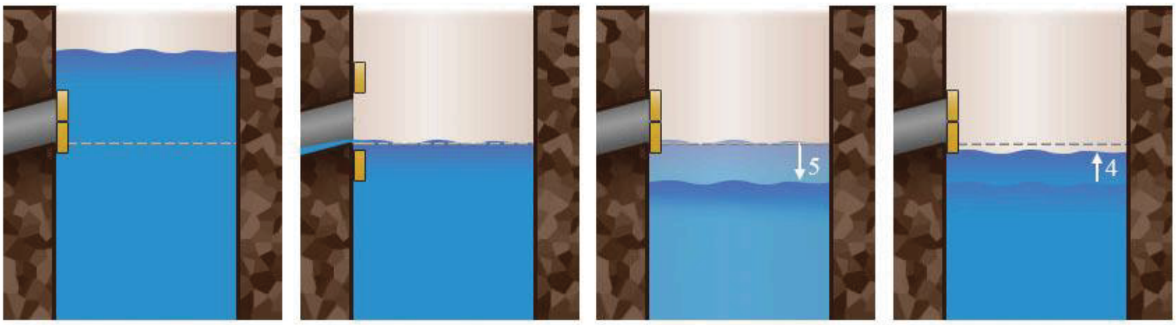
Therefore, the final water level of the reservoir is -11.



For the second scenario, the initial water level of the reservoir is 10.

- The first event is a gate open event. Since $10 > 0$, water flows out from the reservoir and the water level changes to 0.
- After that, a change event with value -5 happens. The water level becomes $0 + (-5) = -5$.
- Finally, another change event with value 4 happens. The water level becomes $-5 + 4 = -1$.

Therefore, the final water level of the reservoir is -1.



2

Input	Output

`2 1`
`2 3`
`1`
	`6`

SUBTASKS

For all cases:
$1 \le N \le 100000$
$1 \le Q \le 100000$
$-10000 \le A_i \le 10000$ for $1 \le i \le N$
$-10^9 \le X_i \le 10^9$ for $1 \le i \le Q$

	Points	Constraints
1	16	$1 \le N \le 1000$ $1 \le Q \le 1000$ $A_i \ne 0$ for $1 \le i \le N$
2	13	$1 \le N \le 1000$ $1 \le Q \le 1000$
3	15	$A_i \ne 0$ for $1 \le i \le N$
4	23	There exists at most 100 different values of $1 \le i \le N$ such that $A_i = 0$
5	21	$A_i \ge 0$ for $1 \le i \le N$
6	12	No additional constraints