TWGSS
OI Team

Cooperation Theory I

INFORMATION

Date	25 February 2023
Time	14:30 - 18:30 (4 hours)

RULES & REMINDERS

- Teams are ordered by the number of **Solved problems**, followed by the **total penalty time**.
- All test cases in a problem must be passed to be considered **Solved** (**Accepted**).
- A submission with the verdict **Wrong Answer**, **Time Limit Exceed** or **Runtime Error** is considered **Rejected**.
- The **penalty time** for a solved problem is the number of minutes past the beginning of the contest that it is solved, plus the number of **Rejected** submissions multiplied by 20 minutes.
- The **total penalty time** is the sum of all solved problems' **penalty time**.
- You are recommended to attempt the problems using C++20. It is not guaranteed that the problems are solvable using other languages.
- Unless otherwise specified, your output will be automatically fixed as follows:
  - Trailing spaces in each line will be removed.
  - An end-of-line character will be added to the end of the output if not present.
- 64-bit integers may be required in some problems, use `long long` in C++ if needed.
- The problems are NOT necessarily arranged in ascending order of difficulty.
- All stories are hypothetically written. In case of coincidences, they are coincidences.

AUTHORS / PREPARED BY

Hsieh Chong Ho

Wong Sheung Chit

Ng Pui Hei

PROBLEMS

		Problem	Limit	
			Time	Memory
TP231	A	Advertisement Master	1s	256MB
	B	Border Definer	1s	256MB
	C	Counting Triangles	1s	256MB
	D	Dim Sum One	1s	256MB
	E	Electronic Strip	1s	256MB
	F	F in the Chat	1s	256MB
	G	Gap by Step	1s	256MB
	H	Heart Arena	1s	256MB
	I	ImPoster Maker	1s	256MB
	J	Junior Cut-off	1s	256MB
	K	King's Gambit	1s	256MB
	L	Long Time No C	1s	256MB

A Advertisement Master

Time Limit 1s
Memory Limit 256MB

"Repair induction cooker (修理電磁爐) 6351xxxx"

You are the *Advertisement Master (廣告達人)*! You recently got a job from an engineer to help him promoting his *induction cooker repairing service*. To have a maximum promotion effect, you decided to write the advertisement everywhere in the city.

There is a total of N rows and M columns of walls in the city. The top-left wall is represented by $(1, 1)$, while the bottom-right wall is represented by (N, M). What you have to do is to write the advertisement on as many walls as possible. Seems easy, right? Unfortunately, no. Police in the city can catch you when you are writing advertisements on the walls!

It is given that a policeman is standing at position (r, c) can "contribute" to position (i, j) with **monitoring level** $\max\{4 - |r - i| - |c - j|, 0\}$. You can only write the advertisement at positions where a policeman is not standing on the same place, and the **monitoring level** is less than or equal to 7.

Apart from being the *Advertisement Master*, you are also the *Coding Master*! Therefore, you decided to write a program to find out the positions that you can write the advertisement on.

INPUT

The first line consists of two integers: N, M, representing the number of rows and columns of walls in the city respectively.

On the i^{th} line of the next N lines, the j^{th} character is `P` if a policeman is standing at that position, and `.` otherwise.

CONSTRAINTS

$1 \leq N, M \leq 2000$

OUTPUT

Output N lines of M characters each.

For each position, output P if a policeman is standing there, output # if it is possible to write the advertisement there, and . otherwise.

SAMPLE TESTS

1

Input	Output
5 7 .... P......	####### P###### ####### ####### #######

The **monitoring levels** are:

```
3  2  1  0  0  0  0
4  3  2  1  0  0  0
3  2  1  0  0  0  0
2  1  0  0  0  0  0
1  0  0  0  0  0  0
```

2

Input	Output
6 7 .... PP..... PP.....	..##### PP.#### PP.#### ..##### ####### #######

The **monitoring levels** are:

```
 8  8  4  1  0  0  0
12 12  8  4  1  0  0
12 12  8  4  1  0  0
 8  8  4  1  0  0  0
 4  4  1  0  0  0  0
 1  1  0  0  0  0  0
```

B Border Definer

Time Limit 1s
Memory Limit 256MB

*"It is difficult to define the border between **good** and **evil** (難為正邪定分界)."*

At the border between **good** and **evil**, the **good** will *work hard to construct (努力興建)*, while the **evil** will *destroy to one's heart's content (盡情破壞)*, turns out *both will suffer (彼此也在捱)*.

The universe can be represented by an infinite grid with each cell having the same size. There is a total of N **evil** people, each of them occupies one cell, while all the other cells are occupied by **good** people.

As the *border definer* of this universe, you don't want both sides to suffer. Therefore, you decide to write a program to determine the cells occupied by the **good** and **evil**, minimising the total length of the border between the two sides.

INPUT

The input consists of an integer: N, representing the number of **evil** people.

CONSTRAINTS

$1 \leq N \leq 10000$

OUTPUT

In the first line, output two space-separated integers: R, C, representing the number of rows and columns of your arrangement respectively.

Each of the following R lines should consist of C characters, showing your arrangement.

For each cell, `#` represents an **evil** person is occupying that cell, while `.` represents a **good** person is occupying that cell.

If there are multiple valid arrangements, output any of them.

REQUIREMENTS

$1 \leq R, C \leq 1000$

The arrangement should consist of exactly N `#`s

SAMPLE TESTS

1

Input	Output
8	2 4 #### ####

The total length of the border is 12, which is minimised.

2

8	3 3 .## ### ###

The total length of the border is 12, which is also a valid output.

3

13	4 5 ####. ####. ####. .#...

The total length of the border is 16, which is minimised.

Please be noted that the extra `.`s are acceptable, as long as the values of R and C fit the requirements.

C Counting Triangles

Time Limit 1s
Memory Limit 256MB

"Circle, triangle, square… I am the best! (圓形三角正方形，樣樣我最醒！)"

Daniel loves circle! He wants to do all things about circle! As a friend of Daniel, you love rectangle! Being your friend, Duncan loves triangle! Duncan wants to do all things about triangle! During a random midnight that Duncan has to go to school the next day, he decided to call you and ask you to write a program to solve a problem about triangle!

Now, you are given N distinct points on a rectangular coordinate system. You are asked to find how many distinct **non-degenerate triangles** can be formed. *(What a boring problem!)*

DEFINITION

A **non-degenerate triangle** means that the three points forming the triangle does not lie on a straight line.

Two triangles are considered different if the one or more points forming the triangles are different.

INPUT

The first line contains an integer: N.

The next N lines each contain two integers: x_i, y_i, representing the coordinates of the i^{th} point.

CONSTRAINTS

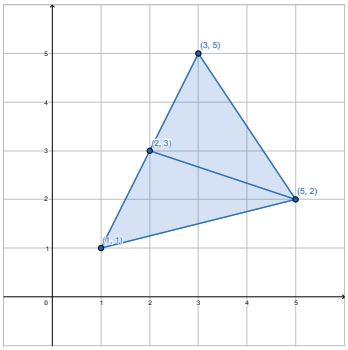
$1 \leq N \leq 500$
$-10^3 \leq x_i, y_i \leq 10^3$
$x_i \neq x_j$ or $y_i \neq y_j$ for all $i \neq j$

OUTPUT

Output a single integer, the number of distinct **non-degenerate triangles** that can be formed.

SAMPLE TESTS

	Input	Output
1	4 1 1 2 3 3 5 5 2	3

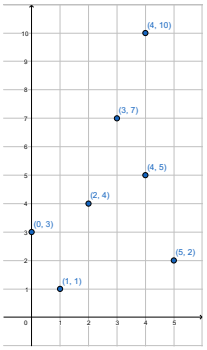


There are in total 3 triangles:

- $(1, 1), (2, 3), (5, 2)$
- $(1, 1), (3, 5), (5, 2)$
- $(2, 3), (3, 5), (5, 2)$

As $(1, 1), (2, 3), (3, 5)$ lie on the same line, so a **non-degenerate triangle** cannot be formed.

2	7 0 3 1 1 2 4 3 7 4 5 4 10 5 2	30



D Dim Sum One

Time Limit 1s
Memory Limit 256MB

You finally got the chance to eat at Hong Kong's hottest new restaurant, Dim Sum One.

After watching the review by *Chris Wong* on the online food guide *Eat & Drink*, you discovered that the dishes are all expensive! But when you realised this, you have already ordered lots of food! Feeling scared having not enough money to pay the bill, you decided to write a program to calculate the total amount you have to pay. Since in case you don't have enough money to pay, you will have to call your mom immediately and ask her for help, or you will need to repay the debt by washing dishes!

You are given the unit prices and the numbers of **small**, **medium**, **large**, **extra-large** dishes you ordered. Note that you also have to pay 10% service charge (加一) on top of the total amount of the dishes. Please find the total amount you have to pay, **round up** to nearest 10 cents ($\$0.1$).

INPUT

The first line contains four numbers with one decimal place: $p_1,\ p_2,\ p_3,\ p_4$, representing the unit price of a **small**, **medium**, **large**, **extra-large** dish respectively.

The second line contains four integers: $a_1,\ a_2,\ a_3,\ a_4$, representing the number of **small**, **medium**, **large**, **extra-large** dishes you have ordered respectively.

CONSTRAINTS

$0 < p_1 < p_2 < p_3 < p_4 \leq 100$
$0 \leq a_1, a_2, a_3, a_4 \leq 100$

OUTPUT

Output the total amount you have to pay, **round up** to nearest 10 cents ($\$0.1$) while **showing one decimal place**.

SAMPLE TESTS

	Input	Output
1	14.5 17.0 20.0 23.0 4 2 3 1	192.5
2	11.1 12.3 13.5 14.7 1 0 0 0	12.3
3	7.0 11.0 13.0 17.0 2 3 2 1	99.0

HINT

The following C++ code snippet can output a floating point number x **rounded off** to (NOT **rounded up** to) and **showing one decimal place**:

```
cout << fixed << setprecision(1) << x;
```

E Electronic Strip

Time Limit 1s
Memory Limit 256MB

Red Cat, An Ox.

You are having Chemistry lesson! As it is the last lesson of form 4, your Chemistry teacher, Miss Chan decided to give the class a sneak peak of the form 5 syllabus. She opened the *Redox Reaction Generator* inside the *Computer Club website*. Suddenly, your ICT teacher, Mr. Lau, walked past the Chemistry Laboratory. When he saw the *Computer Club website* is being projected on the screen, he felt very happy. Therefore, he decided to enter the Chemistry Laboratory and ask you *(Why? Because you study ICT!)* to write a program to solve the following question about redox:

Once upon a time, there is an electronic strip of length N. For each operation, you can choose two cells on the strip as **cathode** (which its charge is positive) and **anode** (which its charge is negative), then you can produce 1 unit of electricity, the charge of the **cathode** will decrease by 1, while the charge of the **anode** will increase by 1.

Now, you are given the arrangement of the electronic strip, what is the maximum units of electricity that can be produced from the strip?

INPUT

The first line contains an integer: N, representing the length of the electronic strip.

The second line contains N integers: $c_1, c_2, \cdots, c_N$. The charge of the i^{th} cell of the strip is c_i. Please be noted that positive and negative charge starts with a $+$ and $-$ sign respectively.

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$
$-10^9 \leq c_i \leq 10^9$

OUTPUT

Output a single integer, the maximum units of electricity that can be produced from the strip.

SAMPLE TESTS

1

Input	Output
3 -1 +2 -3	2

First, by connecting cell c_1 and c_2, 1 unit of electricity can be produced.
The charge of c_1 becomes 0, while the charge of c_2 becomes $+1$.

Next, by connecting cell c_2 and c_3, 1 unit of electricity can be produced.
The charge of c_2 becomes 0, while the charge of c_3 becomes -2.

Now, the arrangement of the strip is $\{0, 0, -2\}$, which no more electricity can be produced.
Therefore, the maximum electricity produced is $1 + 1 = 2$ units.

2

4 +2 0 +2 -3	3

First, by connecting cell c_1 and c_4, 2 units of electricity can be produced.
The charge of c_1 becomes 0, while the charge of c_4 becomes -1.

Next, by connecting cell c_3 and c_4, 1 unit of electricity can be produced.
The charge of c_3 becomes $+1$, while the charge of c_4 becomes 0.

Now, the arrangement of the strip is $\{0, 0, +1, 0\}$, which no more electricity can be produced.
Therefore, the maximum electricity produced is $2 + 1 = 3$ units.

3

6 +1 -2 -3 -5 +7 0	8

F F in the Chat

Time Limit 1s
Memory Limit 256MB

Press F to pay respect.

People always use *the F word* to pay their respect in many situations. When you are told to have a dictation tomorrow, you press *F*. When you are competing in an OI contest without test cases, you press *F*. When you are asked to keep 1.5 metres distance from your classmates, you press *F*.

In a mysterious live stream (broadcasting someone flying to the ceiling), the **respect index** of a message in the chat is the length of the longest **connected** F **component** in it.

You have typed a message of length N in the chat. Suddenly, you saw the streamer successfully fly to the ceiling! In order to pay more respect, you decided to edit your message by replacing no more than K characters in it with F. As you don't know what is the optimal way of replacing characters with F in order to reach the highest **respect index**, you decided to write a program to find a way to do so.

INPUT

The first line consists of two integers: N, K, representing the length of your message and the maximum number of edits that you can make.

The second line consists of a string S of length N, representing your original message.

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$

$0 \leq K \leq N$

S only consists of capital letters

OUTPUT

Output an integer, the maximum **respect index** of the edited message.

SAMPLE TESTS

	Input	Output
1	12 3 FAFBFCFDFEFF	8

By replacing C, D and E in the original message to F, we have FAFBFFFFFFFF, which the length of the longest **connected** F **component** is 8.

2	6 5 FAFAFF	6

3	15 2 CFFCCFFCCCFFFFF	7

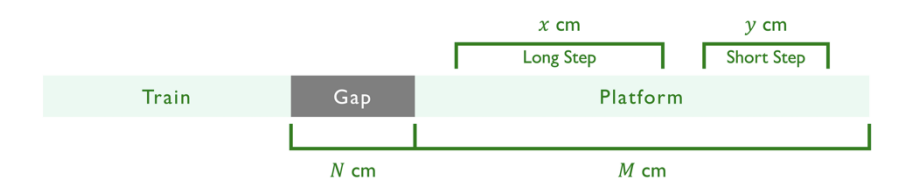
G Gap by Step

Time Limit 1s
Memory Limit 256MB

"Please mind the gap between the train and the platform."

You got 400 marks in HKOI Final Event! Stepping out of the CUHK campus, you entered the University Station. Standing on the platform, a train has come. Suddenly, you discovered that the gap between the train and the platform is too wide! Clearly knowing your own situation and you are now confident of your coding skills (because you've just AKed HKOI), you are going to write a program to determine whether you can safely cross the gap. You will be confident to cross the gap if your code said it is safe, or otherwise you will just leave the station and call a taxi instead. Not wanting to fall into the gap (it hurts!), and you want to save your money (you have to pay for the OI Team dinner as you AKed HKOI), your fate is determined by the program!

The gap between the train and the platform is N cm wide while you are M cm on the platform away from the gap. You can only go forward by x cm (**long step**), or y cm (**short step**) for each move. You can arrange these steps in any combination. Your steps can land on the platform for multiple times, while stepping into the gap means falling into it. Can you make it through the gap safely?



INPUT

The input consists of four integers: N, M, x, y.

CONSTRAINTS

$1 \leq N, M \leq 10^6$

$1 \leq y < x \leq 10^6$

OUTPUT

Output `easy` if you can make it through the gap safely, otherwise, output `taxi`.

SAMPLE TESTS

	Input	Output
1	`6 13 7 5`	`easy`

By taking a **long step** and a **short step**, you have moved $5 + 7 = 12$ cm.
Now, you are 1 cm away from the gap.

By taking a **long step** (7 cm), you successfully cross the total distance of the gap and the distance between the gap and your position ($6 + 1 = 7$ cm).

2	`9 26 10 7`	`taxi`

The furthest distance you can move remain standing on the platform is 24 cm (by taking a **long step** and 2 **short steps**).
Now, you are 2 cm away from the gap.

Even by taking a **long step** (10 cm), you cannot cross the total distance of the gap and the distance between the gap and your position ($9 + 2 = 11$ cm).

3	`5 18 5 2`	`easy`

4	`32 324 35 29`	`taxi`

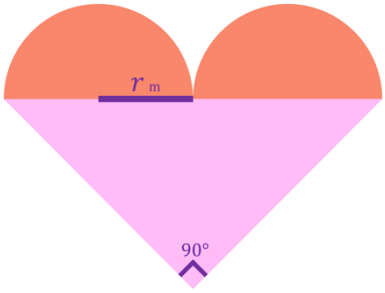
H Heart Arena

Time Limit	1s
Memory Limit	256MB

"Engrossed in competitions~ (投入幾番競技賽～)"

When people are *engrossed in competitions* in *arenas (競技場)*, they will do their best to get the stuff they want. In this *heart-shaped arena*, people are finding love. They know that you are alone, and will probably lose in this competition of chasing Cupid. While they want to know as many details about the arena as possible to give them higher chance to win this important competition.

It is given that the arena consists of two small identical semi-circles with radius r metres, and an isosceles right-angled triangle with the longest side touching the diameters of both semi-circles.



Your task (why it suddenly becomes your task?) is to find the **total area** (in m^2) of the *arena*. Why? Because "*area*" is a subsequence of "*arena*"!

You may ask what is this competition inside the *heart arena*? It's *HKOI Mini Competition 0*!

INPUT

The input consists of an integer: r.

CONSTRAINTS

$1 \leq r \leq 214$

OUTPUT

Output a number, the **total area** of the *heart arena*, **round off** to the nearest 0.001 m^2 while **showing three decimal places**.

SAMPLE TESTS

	Input	Output
1	1	7.142
2	2	28.566
3	14	1399.752
4	38	10312.460

HINT

The following C++ code snippet can directly store the value of π into the variable `PI`:

```
const double PI = acos(-1);
```

In case you don't know how to implement the above code *(but why?)*, the value of π (rounded to 14 decimal places) is: 3.14159265358979.

The following C++ code snippet can output a floating point number `x` **rounded off** to and **showing three decimal places**:

```
cout << fixed << setprecision(3) << x;
```

ImPoster Maker

Time Limit 1s
Memory Limit 256MB

"I'm Poster Maker!"
*"**ImPoster Maker**? You're SUS!"*

You are playing a game called "*Among Us: ImPoster Maker*". Inside the game, you act as the Leading Poster Maker in a group with N Poster Makers. From some reliable sources, you know that there may be an **ImPoster Maker** sneaking inside the group of Poster Makers. A hint specifies that the **ImPoster Maker** is copying your poster design.

A poster with M features can be represented by a string of length M, with each character in the string representing a feature. You check the posters of your group member one by one. The **Imposter Maker** is the one who have the greatest number of common features with your poster design. If there are more than one people having the greatest number of common features with your poster design, the last one you check will be the **Imposter Maker** *(Why? Because that person is unlucky!)*. If everyone in the group has a completely different design from yours, there is no **Imposter Maker** in the group. *(Yay!)*

As the leader of the Poster Makers, and the victim of the pirated copy, you decided to write a program inside the game *(yes, write a program inside the game)* to find out the **ImPoster Maker** in the group according to their posters.

INPUT

The first line consists of two integers: N, M, representing the number of Poster Makers in your group and number of features in a poster respectively.

The second line consists of a string of length M: S_0, representing your poster design.

The i^{th} $(1 \leq i \leq N)$ line of the following N lines each consists of a string of length M: S_i, representing the poster design of the i^{th} person.

CONSTRAINTS

$1 \leq N, M \leq 5000$

S_i only consists of capital letters, where $0 \leq i \leq N$

OUTPUT

If there is no **ImPoster Maker**, output `NOT SUS`.

Otherwise, output an integer, the index of the **ImPoster Maker**.

SAMPLE TESTS

	Input	Output
1	3 6 OITEAM OTTIME OLTEAM OKBEAM	2

For the 1^{st} person, there are 2 common features (character 1 (`O`), 3 (`T`)).
For the 2^{nd} person, there are 5 common features (character 1 (`O`), $3-6$ (`TEAM`)).
For the 3^{rd} person, there are 4 common features (character 1 (`O`), $4-6$ (`EAM`)).

As the 2^{nd} person have the most common features with your design, the 2^{nd} person is the **ImPoster Maker** and `2` should be outputted.

2	3 7 AMONGUS GOODBYE BLUESUS HITWGSS	3

As the 2^{nd} and the 3^{rd} person both have the most (2) common features with your design, the one you check the last is the **ImPoster Maker**.
Therefore the 3^{rd} person is the **ImPoster Maker** and `3` should be outputted.

3	3 7 THEORYI NOTSAME STILLNO NOSLEEP	NOT SUS

All people have no common features with your design, so there is no **ImPoster Maker** and `NOT SUS` should be outputted.

J Junior Cut-off

Time Limit 1s
Memory Limit 256MB

HKOI 2022/23 Junior Group Gold Medal Cut-off Score: 385.6

You couldn't get 400 marks in HKOI Final Event! As you have already stared at the computer screen, waiting for the release of HKOI results for over a week, you felt so nervous about your award. Therefore, you decided to determine the award yourself! Being a Junior Group contestant, you know that all finalists are so kind that they will give you their scores.

There are some rules determining the awards in HKOI:

- A **positive score** is required to be eligible for any awards.
- The minimum score for a **Gold Medal** is the highest score such that at least $1/12$ of all finalists receive a Gold Medal.
- The minimum score for a **Silver Medal** is the highest score such that at least $1/4$ of all finalists receive a Gold or Silver Medal.
- The minimum score for a **Bronze Medal** is the highest score such that at least $1/2$ of all finalists receive a medal.
- For those finalists do not receive medals, the score necessary to achieve an **Honourable Mention** is the highest score such that at least $2/3$ of all finalists in the group receive either a medal or an Honourable Mention.

There is a total of $N+1$ finalists in the Junior Group, given your score and the score of the remaining N finalists. You are going to determine the award that you can get (if any).

INPUT

The first line consists of two integers: N and s_0, representing the number of finalists excluding yourself and your score respectively.

The second line contains N integers: $s_1, s_2, \cdots, s_N$, the score of the i^{th} finalist is represented by s_i.

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$

$0 < s_0 < 400$

$0 \leq s_i \leq 400$, where $1 \leq i \leq N$

OUTPUT

Output a string, representing the award that you would get.

Award	**Gold Medal**	**Silver Medal**	**Bronze Medal**	**Honourable Mention**	No Award
Output	`Gold`	`Silver`	`Bronze`	`HM`	`saf`

It is guaranteed that a solution always exists with the given output.

SAMPLE TESTS

	Input	Output
1	`7 326` `327 294 251 247 170 147 115`	`Silver`
2	`7 294` `327 294 251 247 170 147 115`	`Silver`
3	`13 238` `123 321 145 235 238 237 231 219 205 211 213 162 209`	`Gold`
4	`8 100` `311 251 172 23 11 237 200 100`	`HM`

K King's Gambit

Time Limit 1s
Memory Limit 256MB

"Quickly past the time and seasons, life is like a game of chess, every game is strange and unusual. (倉卒歲月，世事如棋，每局都光怪陸離。)"

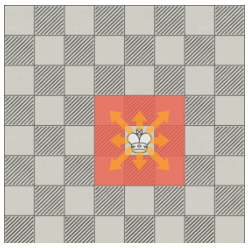
Inspired by the meaningful lyrics, *King Charles* understood that life is just like a game of chess. To be a successful King, he decided to learn how to play chess. He asked the top computer scientists to craft a chess program for him to practice. Being one of the computer scientists, you wanted to impress *King Charles* by developing an original mode.

The chess rules are being simplified to lower the difficulty of programming the game mode, they are not guaranteed to be valid in a real chess game.

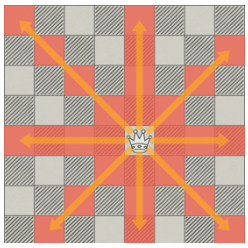
Here are the positions of enemy piece that different pieces can capture:

King	exactly one square horizontally, vertically, or diagonally
Queen	horizontally, vertically or diagonally
Rook	horizontally or vertically
Knight	two squares horizontally and one square vertically, or one square horizontally and two squares vertically ("L" pattern)
Bishop	diagonally
Pawn	nearest square diagonally in front of it (upward for **white**, downward for **black**)

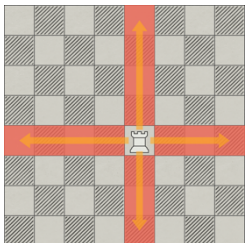
King



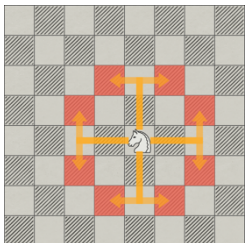
Queen



Rook

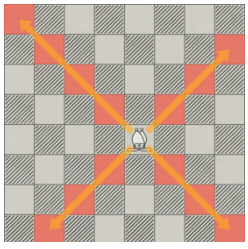


Knight

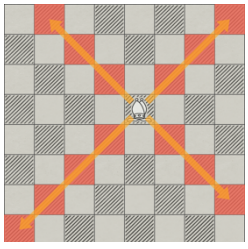


Bishop

(light-squared)

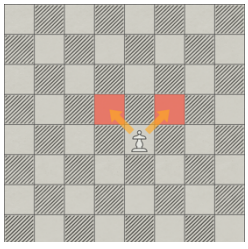


(dark-squared)

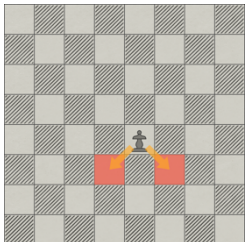


Pawn

(white)



(black)



The **King** is in **check** when it can be captured by any enemy piece.

Here are the **material points** of different pieces:

King	no points
Queen	9 points
Rook	5 points each
Knight	3 points each
Bishop	
Pawn	1 point each

The objective of the mode is simple: given the **material points** of both sides (**white** and **black**), the player has to construct a valid arrangement of the pieces on the chess board.

A valid arrangement of the pieces on the chess board has the following properties:

- There must be a **white King** and a **black King** on the board.
- Two **Kings** CANNOT be BOTH in **check**. (It is reminded that white pawns are moving upward and black pawns are moving downward.)
- For **each side**, there are **at most**:
  - **1 Queen**
  - **2 Rooks**
  - **2 Knights**
  - **1 light-squared Bishop**
  - **1 dark-squared Bishop**
  - **8 Pawns**

You didn't want *King Charles* to encounter any bugs when playing the game, so you decided to playtest the game mode. But you were too lazy to construct the board arrangements manually, so you decided to write a program to do so.

INPUT

The input consists of two integers: P_w and P_b, representing the **material points** of **white** side and **black** side respectively.

CONSTRAINTS

$0 \leq P_w, P_b \leq 39$

OUTPUT

Output a valid arrangement of the pieces on the chess board. The output should consists of 8 lines, each containing 8 strings of length 2, each separated by a space.

Use `--` to indicate empty square and use chess notations to indicate the piece on the square.
The piece should be represented in the following notations:

	King	Queen	Rook	Bishop	Knight	Pawn
White	`Kw`	`Qw`	`Rw`	`Bw`	`Nw`	`Pw`
Black	`Kb`	`Qb`	`Rb`	`Bb`	`Nb`	`Pb`

If there are multiple valid arrangements, output any one of them.

SAMPLE TESTS

1

Input	Output

`0 0`	`Kb -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- Kw`

This is a **valid** arrangement.

2

`3 0`	`Kb -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- Nw -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- --`
	`-- -- -- -- -- -- -- Kw`

This is a **valid** arrangement.

3

```
3 0
```

```
-- -- -- -- -- -- -- --
-- -- -- -- Kb -- -- --
-- -- -- -- -- -- -- --
-- -- Pw Pw Pw -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- Kw
```

This is a **valid** arrangement.

4

```
3 3
```

```
-- -- -- -- -- -- -- --
-- -- -- -- Kb -- -- --
-- -- -- -- -- -- -- --
-- -- -- Nw -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- Nb -- --
-- -- -- -- -- -- -- Kw
```

This is an **INVALID** arrangement.

Reason: both **Kings** are in **check**

5

```
6 0
```

```
-- -- -- -- -- -- -- --
-- -- -- -- Kb -- -- --
-- -- -- -- -- -- -- --
-- -- -- Bw -- -- -- --
-- -- -- -- Bw -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- Kw
```

This is an **INVALID** arrangement.

Reason: 2 same-coloured **Bishops** for **white** side

L Long Time No C

Time Limit 1s
Memory Limit 256MB

#456covidweare #Cstandsfor6C

Class 6C has the name of "*COVID (Cute Optimistic Victorious Intelligent Democratic) class*" as students in this class, sadly, caught COVID. When a student caught COVID, he/she have to start his/her quarantine period immediately, that day is the 0^{th} day of the quarantine period. The quarantine period ends on the 7^{th} day. (i.e. only the 0^{th} day to the 6^{th} day count as under quarantine)

When more than or equal to 5 students are under quarantine on the same day, the class have to **study from home**, preventing students from other classes having nightmares when they see a 6C student walking in the school campus.

Ending the school year, Jack, a student from 6C, decides to calculate the total numbers of days that his class was **studying from home (SFH)** in the past year. He gets the record of the date that each student caught COVID, but he doesn't want to spend the time calculating the value by hand. As an "*old ghost*" of OI Team, he asks you, an OI Team member, to write a program, helping him to calculate the value he wants. *(Why doesn't he code himself? Such a lazy person!)*

INPUT

The first line consists of an integer: N, representing the number of students who have caught COVID in the past year.

The second line contains N integers: $d_1, d_2, \cdots, d_N$.
The i^{th} person caught COVID on the $d_i{}^{\text{th}}$ school day of the school year. As Class 6C is in a *parallel universe*, a school year may have more than 366 days. Please be noted that d_i are NOT necessarily sorted in ascending order.

CONSTRAINTS

$1 \leq N \leq 2 \times 10^5$
$1 \leq d_i \leq 10^9$

OUTPUT

Output an integer, the total number of days that his class was **studying from home (SFH)**.

SAMPLE TESTS

	Input	Output
1	10 1 1 1 1 2 5 7 7 9 10	8

Day	New Cases (+)	Ends Quarantine (-)	Under Quarantine	SFH
0			0	×
1	4		4	
2	1		**5**	**6 days**
3				
4				
5	1		**6**	
6				
7	2		**8**	
8		4 (day 1 cases)	4	×
9	1	1 (day 2 case)		
10	1		**5**	**2 days**
11				
12		1 (day 5 case)	4	×
13				
14		2 (day 7 cases)	2	
15				
16		1 (day 9 case)	1	
17		1 (day 10 case)	0	
		Total		**8 days**

2	9 13 15 17 12 14 16 18 11 10	7

3	20 22 13 31 23 14 33 4 7 5 31 13 13 33 42 34 14 16 18 21 13	11