TWGSS OI Team

2023/24 Team Selection Test

SCHEDULE

Date	19 April 2024
Time	16:05 - 18:20 | 2 hours 15 minutes

PROBLEMS

	Problem	Limit		Score	
		Time	*Memory*	*Subtasks*	*Total*
TS240	Relaxing Sunday	1s	256MB	5 | 7 | 8	20
TS241	Character Repetition	1s	256MB	14 | 4 | 8 | 4	30
TS242	Fuk Luk Sau	1s	256MB	6 | 12 | 6 | 10 | 6	40
TS243	Grand School Award	1s	256MB	12 | 20 | 15 | 10 | 8	65
TS244	Tiers of the Kingdom	1s	256MB	9 | 15 | 13 | 16 | 18 | 24	95
				Full Score	250

REMINDERS

- All problems are divided into subtasks. All test cases in a subtask must be passed to get points.
- You may attempt the problems using C++20 or Python 3 (secondary language). It is not guaranteed that the problems are solvable using secondary language.
- 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.
- All stories are hypothetically written. In case of coincidences, they are coincidences.

TS240 Relaxing Sunday

Score	Limit
20	1s | 256MB

You know, today is Sunday, I would like to relax.

Mr. Elder is tired of working endlessly every day! As he is no longer a young person, he decides that he would take rest on Sundays.

As the most loyal fan of Mr. Elder, you decide to write a program to determine whether the d^{th} day of the year with the first day being S is **Sunday**.

INPUT

The first line consists of a string S, the day of the week of the first day on the queried year.

The second line consists of an integer d, the queried day in the year.

OUTPUT

If the d^{th} day of the queried year is **Sunday**, output `Relax`.

Otherwise, output `Angry`.

SAMPLE TESTS

	Input	Output
1	`Monday` `1`	`Angry`

This sample satisfies the constraints of Subtask 1.

	Input	Output
2	`Sunday` `15`	`Relax`

This sample satisfies the constraints of Subtask 2.

	Input	Output
3	`Saturday` `53`	`Angry`
4	`Wednesday` `222`	`Relax`

CONSTRAINTS

$S =$ `Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday` or `Saturday`

$1 \leq d \leq 365$

SUBTASKS

	Points	Constraints
1	5	$d = 1$
2	7	$S =$ `Sunday`
3	8	No additional constraints

TS241 Character Repetition

Score	Limit	
30	1s	256MB

"thank youuuu~"
"heyyyyyyy!"
"okkkkk~"

What is going on?!?! Our messages are flooded with repeated consecutive characters! They're horrible as they not only do no help with any conversations, but they also take up many unnecessary storages!

Of course, you know that saying this fact out is hurting many brittle glass hearts who like sending those terrible things out. Therefore, you decide to write a program to simplify a string by removing repeated consecutive characters locally on your device! Formally, given a string S, for every $1 \leq i \leq N-1$ if $S_i = S_{i-1}$, your program should remove S_{i+1} in the resultant string. Note that the comparation is **case-insensitive** (uppercase and lowercase of an English letter is considered the same).

INPUT

The first and only line consists of a string S to be processed.

OUTPUT

Output the simplified string after removing all repeated consecutive characters in S.

SAMPLE TESTS

	Input	Output
1	`halllooo`	`halo`

This sample satisfies the constraints of Subtask 1.

	Input	Output
2	`hmmm many spaces`	`hm many spaces`

This sample satisfies the constraints of Subtask 2.

	Input	Output
3	`PplzzZzZzZz`	`Plz`

This sample satisfies the constraints of Subtask 3.

	Input	Output
4	`TWGSS OI Team is good`	`TWGS OI Team is god`

CONSTRAINTS

$1 \leq \text{length of } S \leq 1000$

S consists of English letters (`A` - `Z`, `a` - `z`) and spaces only

SUBTASKS

	Points	Constraints
1	14	S consists of lowercase English letters (`a` - `z`) only
2	4	S consists of lowercase English letters (`a` - `z`) and spaces only
3	8	S consists of English letters (`A` - `Z`, `a` - `z`) only
4	4	No additional constraints

TS242 Fuk Luk Sau

Score	Limit
40	1s | 256MB

The annual "Fuk Luk Sau" sale of electrical appliances finally arrives! As usual, the electrical appliances are divided into 3 categories – Fuk (福, lucky), Luk (祿, wealthy) and Sau (壽, healthy). They are sold in packages, which a package consists of an item in each category.

There is a total of N_F Fuk items, N_L Luk items and N_S Sau items. For the i^{th} Fuk, Luk and Sau item, the original price is $\$F_i$, $\$L_i$ and $\$S_i$ respectively. The Fuk item chosen determines the package price, which is P_i if the i^{th} Fuk item is chosen. The Luk and Sau item chosen would not affect the package price. The discount of the package is calculated by the sum of original prices of the 3 items chosen minus the package price. Note that it is possible to have negative discount, i.e. it becomes more expensive to buy the package than buying the items separately!

"Refund! Refund! Refund!" You've already spent too much money to a disappointing football match. Therefore you decide to buy a Fuk Luk Sau package with the combination of items having the highest discount. Note that if all package combinations have negative discounts, you decide not to buy the package in order to have the discount of $\$0$. As an honourable programmer, you decide to write a program to find out the maximum discount.

INPUT

The first line consists of three integers N_F, N_L and N_S, representing the number of Fuk, Luk ans Sau items respectively.

The second line consists of N_F integers, the i^{th} integer P_i represents the package price if the i^{th} Fuk item is chosen.

The third line consists of N_F integers, the i^{th} integer F_i represents the original price of the i^{th} Fuk item.

The fourth line consists of N_L integers, the i^{th} integer L_i represents the original price of the i^{th} Luk item.

The fifth line consists of N_S integers, the i^{th} integer S_i represents the original price of the i^{th} Sau item.

OUTPUT

Output an integer, the maximum discount.

SAMPLE TESTS

	Input	Output
1	1 1 1 64 50 20 19	25

This sample satisfies the constraints of Subtask 1.

	Input	Output
2	4 3 2 19 61 20 21 6 48 9 19 9 5 7 8 6	15

	Input	Output
3	4 3 2 10 20 30 40 1 12 23 34 3 2 1 2 1	0

CONSTRAINTS

$1 \leq N_F, N_L, N_S \leq 2 \times 10^5$

$1 \leq P_i, F_i, L_i, S_i \leq 10^9$

SUBTASKS

	Points	Constraints
1	6	$N_F = N_L = N_S = 1$ $1 \leq P_i, F_i, L_i, S_i \leq 5000$
2	12	$1 \leq N_F, N_L, N_S \leq 20$ $1 \leq P_i, F_i, L_i, S_i \leq 5000$
3	6	$1 \leq N_F, N_L, N_S \leq 2000$ $1 \leq P_i, F_i, L_i, S_i \leq 5000$
4	10	$1 \leq P_i, F_i, L_i, S_i \leq 5000$
5	6	No additional constraints

TS243 Grand School Award

Score	Limit	
65	1s	256MB

"Take the plunge without hesitation, Grand School Award will no longer be our imagination! (義無反顧搏 1 鋪，Grand School Award 我攞到！)"

Despite the award name is "Grand School Prize" instead of "Award", members of *Team Tsuen Wan* had paid their best efforts in the this year's HKOI. Sadly, the results are worse than expectations. Members are asked to write a reflection reviewing their performance in the competition. At the same time, the trainers are busy analysing the data of previous competitions to find out the change in ranks of schools over the years.

They intend to develop a software, which will works as follows:

- At first, N data entries describing the performance of schools over the years will be inputted. The i^{th} entry will consist of the school name S_i, the competition year Y_i and the points got by the school in that year P_i. It is guaranteed that there is no repeated entries, i.e. no two entries have the same S_i and Y_i.
- After entering all data, Q queries will be performed regarding the ranks of the schools. The j^{th} query will have two parameters, school name s_j and competition year y_j.
  - If there is no data entry found for the school s_j in the competition year y_j, the return value should be -1.
  - Otherwise, the rank of the school s_j in the competition year y_j should be returned. Note that the rank is ordered by high to low of the points, with the school having the highest points in that year having the rank 1. It is also guaranteed that for each year, no two schools have the same points.

Although the trainers are obviously capable of programming such software in minutes, they still decide to ask you to code it for them *as a practice*, how unfair!

INPUT

The first line consists of two integers N and Q, representing the number of data entries and the number of queries respectively.

The i^{th} line of the next N line describes the i^{th} data entry, which consists of a string S_i and two integers Y_i and P_i, representing the school name, competition year and its points respectively.

The j^{th} line of the next Q line describes the j^{th} query, which consists of a string s_j and an integer y_j, representing the queried school name and the competition year respectively.

OUTPUT

Output Q lines.

On the j^{th} line, output an integer, the return value of the j^{th} query.

SAMPLE TESTS

1

Input	Output
6 1 TWGSS 2023 14 PCMS 2023 33 DBS 2023 43 WYK 2023 22 LSC 2023 18 CIS 2023 9 TWGSS 2023	5

This sample satisfies the constraints of Subtask 1.

2

Input	Output
5 6 A 1111 10 B 2222 10 A 2222 30 C 1111 20 C 2222 20 A 1111 B 1111 C 1111 A 2222 B 2222 C 2222	2 -1 1 1 3 2

CONSTRAINTS

$1 \le N, Q \le 2 \times 10^5$

$1 \le$ length of $S_i, s_j \le 5$

S_i and s_j consists of uppercase English letters (`A` - `Z`) only

$1 \le Y_i, P_i, y_j \le 10^9$

$S_a \ne S_b$ or $Y_a \ne Y_b$ for $a \ne b$

SUBTASKS

	Points	Constraints
1	12	$1 \le N, P_i \le 5000$ $Q = 1$ $S_x = s_1$ for exactly one x $Y_1 = Y_2 = \cdots = Y_N = y_1$
2	20	$Y_1 = Y_2 = \cdots = Y_N = y_1 = y_2 = \cdots = y_Q$
3	15	$1 \le N, Q, Y_i, P_i, y_j \le 5000$
4	10	$1 \le N, Q \le 5000$
5	8	No additional constraints

TS244 Tiers of the Kingdom

Score	Limit	
95	1s	256MB

On the sky of the *Kingdom of Hirule*, there are N tiers of floating islands. A tier i island has the power level P_i. As a brave and smart adventurer *Linkg*, you can use your magic to combine K tier i ($1 \leq i \leq N-1$) islands into one tier $i+1$ island.

Looking around, you discover that there are A_i tier i islands. Wanting to rescue *Princess Cellda* as soon as possible, you decide to use your strong coding skills to write a program in calculating the maximum total power level of the islands by combining them optimally.

INPUT

The first line consists of two integers N and K, representing the numbers of tiers for islands and the number of islands required combining to an island of the next level respectively.

The second line consists of N integers, the i^{th} integer P_i represents the power level of a tier i island.

The third line consists of N integers, the i^{th} integer A_i represents the number of tier i island at the beginning.

OUTPUT

Output an integer, the maximum total power level of the islands by combining them optimally.

SAMPLE TESTS

	Input	Output
1	3 2 1 3 9 7 1 2	37

- Combining 6 tier 1 islands into 3 tier 2 islands.
- Combining 4 tier 2 islands into 2 tier 3 islands.

There are 1 tier 1 island and 3 tier 3 islands at the end.

2	3 3 1 3 9 7 1 2	28

3	7 2 1 2 3 10 4 6 1 7 5 1 1 2 0 4	47

Note that it is not necessary that $P_i < P_{i+1}$.

CONSTRAINTS

$1 \leq N \leq 5000$

$2 \leq K \leq 10^7$

$0 \leq P_i, A_i \leq 10^7$

SUBTASKS

	Points	Constraints
1	9	$N = 2$
2	15	$N = 3$
3	13	$A_1 = K$ $0 \leq A_i < K$ for $1 < i \leq N$
4	16	$1 \leq N \leq 10$
5	18	$0 \leq A_i \leq K$
6	24	No additional constraints