

NSSPC Final 2024

National Taiwan University

2024/09/16

Language	Version	Compile Flags	Extensions
C	gcc 11.3.0	-O2 -x c -static -lm	.c
C++	g++ 11.3.0	-O2 -std=gnu++20 -static	.cc, .cpp, .cxx, .c++
Java	openjdk 17.0.12	[Omitted]	.java

Problem	Problem Name	Points	Time Limit	Memory Limit
A	Nice NSSPC Final	30	1 s	512 MB
B	Waktu Kerja	5	1 s	512 MB
C	Liga Besbol	10	1 s	512 MB
D	Rentetan Dalam Rentetan	15	1 s	512 MB
E	Penggredan	15	1 s	512 MB
F	Penguji	15	1 s	512 MB
G	Star Battle Puzzle	15	1 s	512 MB
H	Membaiki Istana	15	1 s	512 MB
I	Operasi Jadual	20	1 s	512 MB
J	Pokok yang Membesar	20	1 s	512 MB
K	Kiub Rubik	30	1 s	512 MB
L	Pembuat Obor	30	1 s	512 MB
M	Data Eksperimen	30	1 s	512 MB
N	Kemaskan Desktop	40	1 s	512 MB
O	Batang Mancis Ai-chan	40	1 s	512 MB



This page is intentionally left blank.

A. Nice NSSPC Final

Problem ID: nice



Tahniah kerana mara ke peringkat akhir NSSPC 2024! Mari kita bermula dengan mencetak satu ayat: “Nice, NSSPC 2024 Final!”.

Input

Tiada input untuk soalan ini.

Output

Output satu ayat Nice, NSSPC 2024 Final!.

Sample Input 1

Sample Output 1

	Nice, NSSPC 2024 Final!
--	-------------------------



This page is intentionally left blank.

B. Waktu Kerja

Problem ID: work



WiwiHo mementingkan pengurusan masa. Dia suka bekerja selama X jam berterusan dan kemudiannya berehat selama Y jam. Dia akan sambung bekerja sejeurus selepas berehat. Contohnya, jika $X = 5$ dan $Y = 2$, maka WiwiHo akan bekerja selama 5 jam berterusan dan kemudiannya berehat selama 2 jam.

Andaikan dia mula bekerja dari permulaan, berapakah jumlah jam yang WiwiHo akan bekerja dalam tempoh Z jam?

Input

Input hanyalah satu baris dengan tiga integer, X , Y dan Z .

- $1 \leq X, Y \leq 100$
- $1 \leq Z \leq 10^9$

Output

Satu integer iaitu jumlah jam yang WiwiHo akan bekerja.

Sample Input 1

5 2 20

Sample Output 1

15

Sample Input 2

99 1 24

Sample Output 2

24



This page is intentionally left blank.

C. Liga Besbol

Problem ID: league



Di negara Keledok, liga besbol diadakan setiap tahun dari bulan Mac hingga Oktober. Terdapat enam pasukan dalam liga, diwakili oleh huruf besar **A, B, C, D, E, F**. Liga berlangsung melalui satu siri N acara, dan terdapat dua jenis acara:

- Acara perlawanan: Dua pasukan yang berbeza bermain dalam satu perlawanan besbol, dan salah satu daripada mereka menang.
- Acara pemberian mata: Pasukan dengan kadar kemenangan tertinggi pada masa itu akan menerima 1 mata. Jika terdapat lebih daripada satu pasukan dengan kadar kemenangan yang sama, maka **tiada pasukan yang akan menerima mata**. Kadar kemenangan dikira sebagai bilangan kemenangan dibahagikan dengan jumlah perlawanan yang dimainkan oleh pasukan tersebut. Dijamin bahawa sekurang-kurangnya satu perlawanan telah dimainkan oleh setiap enam pasukan apabila acara ini berlaku.

Diberi semua acara dalam urutan kronologi, kira berapa banyak mata yang telah diperoleh oleh setiap pasukan.

Input

Baris pertama input mengandungi satu integer positif N , yang mewakili bilangan acara.

N baris seterusnya menerangkan acara dalam urutan yang berlaku.

Jika acara ke- i adalah acara perlawanan, baris ke- i akan mengandungi tiga rentetan t_i, x_i, y_i di mana $t_i = \text{match}$, dan x_i, y_i adalah pasukan, dengan x_i sebagai pasukan yang menang.

Jika acara ke- i adalah acara pemberian mata, baris ke- i akan mengandungi satu rentetan t_i di mana $t_i = \text{point}$.

- $1 \leq N \leq 1000$
- $t_i \in \{\text{match}, \text{point}\}$
- $x_i, y_i \in \{\text{A, B, C, D, E, F}\}$
- $x_i \neq y_i$



Output

Keluarkan satu baris dengan 6 integer, yang mewakili mata untuk A, B, C, D, E, F mengikut urutan.

Sample Input 1	Sample Output 1
11 match A D match B E match C F point match A B match B C point match C A point match B E point	1 1 0 0 0 0
Sample Input 2	Sample Output 2
13 match A C match B D match A E match B F point match A D match B C point match A B match B A point match A D point	1 0 0 0 0 0

D. Rentetan Dalam Rentetan

Problem ID: strings



Sam memiliki dua rentetan s dan t yang terdiri daripada huruf kecil Inggeris sahaja. Dia ingin tahu sama ada dua rentetan itu serupa. Dalam definisinya, dua rentetan adalah serupa jika mereka mempunyai satu rentetan biasa.

Oleh kerana rentetan-rentetan tersebut boleh menjadi agak panjang, Sam meminta bantuan anda untuk mencari sebarang rentetan biasa dalam s dan t .

Rentetan b adalah sebahagian rentetan a jika b muncul secara bersambung dalam a . Rentetan c adalah rentetan biasa a dan b jika c adalah sebahagian rentetan a dan sebahagian rentetan b .

Input

Input terdiri daripada dua baris. Baris pertama terdiri daripada rentetan s dan baris kedua terdiri daripada rentetan t .

- $1 \leq |s| \leq 10^6$, di mana $|s|$ mewakili panjang s .
- $1 \leq |t| \leq 10^6$, di mana $|t|$ mewakili panjang t .
- Semua aksara ialah huruf kecil Inggeris (**a** hingga **z**.)

Output

Jika rentetan s dan t tidak mempunyai sebarang rentetan biasa, keluarkan “NO” (tanpa tanda petikan) dalam satu baris.

Jika tidak, keluarkan dua baris. Pada baris pertama, keluarkan “YES” (tanpa tanda petikan). Pada baris kedua, keluarkan mana-mana rentetan biasa antara s dan t . Jika terdapat lebih daripada satu rentetan biasa, keluarkan mana-mana satu daripadanya.

Sample Input 1	Sample Output 1
looks good	YES oo



Sample Input 2

Sample Output 2

hello nsspc	NO
----------------	----

Sample Input 3

Sample Output 3

abcdefghijklmnopqrstuvwxy abcdefghijklmnopqrstuvwxy	YES abcdefghijklmnopqrstuvwxy
--	----------------------------------

E. Penggredan

Problem ID: grading



Profesor Li menawarkan kursus yang dipanggil “Struktur Data Selang Lanjutan”. Memandangkan tajuk kursus mengandungi perkataan “Lanjutan”, ia bukanlah kursus yang boleh didaftarkan dengan mudah oleh pelajar; mereka mesti lulus ujian daripada Profesor Li untuk layak.

Untuk memilih pelajar yang berkebolehan mengikuti kursus ini, Profesor Li telah merangka satu tugas. Jawapan yang betul untuk tugas ini terdiri daripada rentetan N : s_1, s_2, \dots, s_N . Setiap rentetan s_i terdiri daripada huruf besar dan huruf kecil Inggeris dan mempunyai panjang antara 1 hingga 20. Percubaan pelajar akan dianggap betul jika mereka menghantar rentetan N ini kepada Profesor Li dalam **sebarang susunan**. Jika tidak, jawapan itu akan dianggap tidak betul.

Pelajar hanya tahu bahawa jawapan yang betul untuk masalah ini terdiri daripada 1 hingga 20 rentetan huruf besar dan huruf kecil Inggeris, di mana setiap rentetan mempunyai panjang antara 1 hingga 20. Secara formal, percubaan pelajar akan memasukkan integer M , yang mewakili bilangan rentetan yang mereka percaya adalah jawapannya, serta M rentetan t_1, t_2, \dots, t_M , yang mewakili rentetan dalam percubaan mereka. Percubaan mereka akan dianggap betul jika $N = M$ dan wujud satu pilih atur p_1, p_2, \dots, p_N daripada 1 hingga N , supaya $\forall 1 \leq i \leq N, s_i = t_{p_i}$. Dalam erti kata lain, ini bermakna bahawa t_1, t_2, \dots, t_M boleh disusun semula menjadi s_1, s_2, \dots, s_N . Pilih atur 1 kepada N ditakrifkan sebagai urutan di mana setiap $1, 2, \dots, N$ muncul tepat sekali.

Walaupun kursus ini sangat mencabar, ramai pelajar masih ingin mendaftar. Terdapat pelajar T yang telah cuba menyelesaikan masalah ini, dan anda perlu memberitahu Profesor Li sama ada jawapan setiap pelajar adalah betul.

Input

Baris pertama mengandungi integer N yang mewakili bilangan rentetan dalam jawapan yang betul.

Baris kedua mengandungi N rentetan s_1, s_2, \dots, s_N yang mewakili rentetan N dalam jawapan yang betul, dipisahkan oleh ruang.

Baris ketiga mengandungi integer T yang mewakili bilangan pelajar yang telah menyerahkan jawapan mereka.

T barisan yang seterusnya, di mana barisan ke- i mewakili penyerahan pelajar ke- i , setiap

satu mengandungi integer M yang mewakili bilangan rentetan yang pelajar percaya ada dalam jawapan, diikuti dengan M rentetan t_1, t_2, \dots, t_M , dipisahkan oleh ruang.



- $1 \leq N, M \leq 20$
- $1 \leq T \leq 10000$
- Setiap s_i dan t_i ialah rentetan huruf besar dan huruf kecil Inggeris dengan panjang di antara 1 hingga 20.

Output

Output T baris, di mana baris ke- i mengandungi **Yes** jika jawapan murid ke- i adalah betul, jika tidak, output **No**.

Sample Input 1	Sample Output 1
5 good luck and have fun	No
4	Yes
3 luck and fun	No
5 have fun and good luck	Yes
6 good luck and and have fun	
5 good luck and have fun	

Sample Input 2	Sample Output 2
3	No
Welcome to NSSPC	Yes
5	No
3 WELCOME TO NSSPC	No
3 to Welcome NSSPC	No
2 Welcome NSSPC	
1 Welcome	
3 welcome to nsspc	

Sample Input 3

Sample Output 3



<p>6 to be or not to be 3 4 to be or not 6 to to be be or not 9 to be or not to be or to be</p>	<p>No Yes No</p>
---	--------------------------



This page is intentionally left blank.

F. Penguji

Problem ID: testers



NSSPC 2024 semakin hampir, dan para hakim mesti menyiapkan soalan secepat mungkin. Terdapat N orang hakim dalam pasukan penghakiman, dan mereka merancang untuk menyediakan sejumlah N soalan, bernombor dari 1 hingga N . Pengarang soalan i ialah hakim i .

Untuk memastikan kualiti soalan tersebut, setiap soalan mesti diuji oleh hakim selain daripada pengarang untuk mengesahkan bahawa tiada isu yang berlaku. Khususnya, setiap hakim mesti bertanggungjawab untuk menguji satu soalan sahaja, dan setiap soalan mesti mempunyai tepat satu hakim yang bertanggungjawab untuk mengujinya. Hakim yang bertanggungjawab menguji tidak boleh menjadi pengarang soalan tersebut. Beberapa hakim sudah memilih soalan yang akan mereka uji. Oleh kerana tarikh akhir semakin dekat, anda mesti cepat menugaskan baki hakim kepada soalan yang akan mereka bertanggungjawab untuk menguji.

Input

Baris pertama mengandungi satu integer T , mewakili bilangan kes ujian.

Untuk setiap kes ujian, baris pertama mengandungi satu integer N , dimana ia mewakili bilangan hakim, dan juga bilangan soalan yang perlu disediakan.

Baris kedua mengandungi N integer a_1, a_2, \dots, a_N . Untuk $1 \leq i \leq N$, jika $a_i \neq -1$, ini bermakna hakim i telah memilih untuk menguji soalan a_i ; jika $a_i = -1$, ini bermakna hakim i belum memilih soalan mana yang beliau akan uji.

- $1 \leq T$
- $2 \leq N \leq 10^5$
- Jumlah keseluruhan N dalam semua kes ujian $T \leq 10^5$
- $1 \leq a_i \leq N$ atau $a_i = -1$
- Ia dijamin bahawa ada jawapan untuk menugaskan soalan kepada semua hakim dengan $a_i = -1$ dengan semua syarat yang tertera atas dipenuhi



Output

Untuk setiap kes ujian, output adalah satu baris yang mengandungi N integer b_1, b_2, \dots, b_N , di mana $1 \leq b_i \leq N$, yang mewakili soalan yang akan bertanggungjawab diuji oleh hakim i selepas penugasan dilakukan. Jika seorang hakim telah memilih soalan untuk diuji, anda mesti mengekalkan pilihan asal mereka. Secara formal, output anda mesti memenuhi semua syarat berikut:

- $1 \leq b_i \leq N$
- $b_i \neq i$
- Jika $a_i \neq -1$, maka $b_i = a_i$
- Untuk $1 \leq j \leq N$, hanya satu i di mana $b_i = j$

Sample Input 1

Sample Input 1	Sample Output 1
5	2 3 4 5 1
5	3 4 2 5 1
-1 -1 -1 -1 -1	3 1 2
5	2 3 1
3 4 2 5 1	3 8 10 5 2 9 1 6 7 4
3	
3 -1 -1	
3	
-1 -1 1	
10	
3 -1 -1 5 2 -1 1 6 -1 4	

G. Star Battle Puzzle

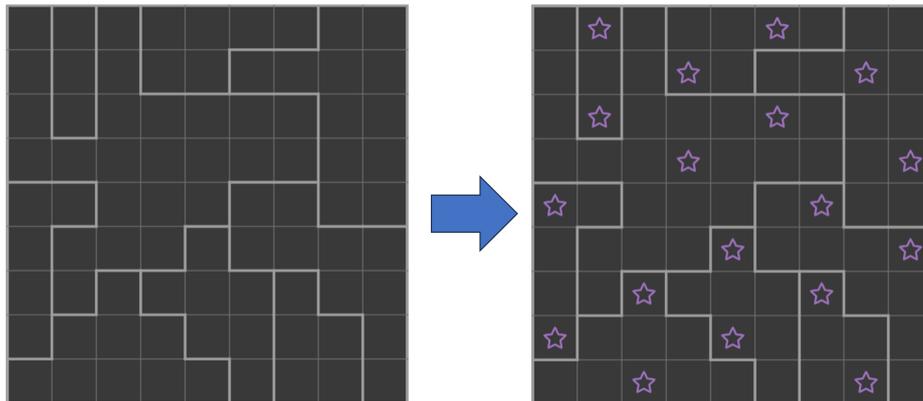
Problem ID: star-battle



Pernahkah anda bermain permainan Star Battle Puzzle? Star Battle Puzzle adalah permainan teka-teki logik, di mana setiap teka-teki bermula dengan papan yang kosong. Papan yang kosong ini mempunyai 9×9 sel yang dibahagikan kepada beberapa kawasan. Matlamat pemain adalah untuk meletakkan “bintang” ke dalam sel-sel tersebut, dengan memenuhi syarat-syarat berikut:

- Setiap baris mesti mengandungi tepat dua bintang.
- Setiap lajur mesti mengandungi tepat dua bintang.
- Setiap kawasan mesti mengandungi tepat dua bintang.
- Tiada dua bintang boleh bersebelahan dalam mana-mana lapan arah. Ini bermaksud, sel-sel yang mengandungi bintang tidak berkongsi sebarang tepi atau sudut.

Rajah di bawah menunjukkan contoh papan awal dan penyelesaiannya:



Sekarang, sila tulis satu program yang diberikan Star Battle Puzzle dengan bintang yang telah diletakkan, program anda perlu menentukan sama ada Star Battle Puzzle itu diselesaikan dengan betul atau tidak. Maksudnya, sama ada bintang yang diletakkan dalam sel-sel tersebut memenuhi semua syarat yang ditetapkan.

Input

Input bermula dengan maklumat papan awal, terdiri daripada sembilan baris, setiap satu dengan sembilan aksara, di mana setiap aksara adalah digit dari 1 hingga 9, mewakili nombor kawasan bagi sel tersebut.



Ini diikuti dengan penempatan bintang, terdiri daripada sembilan baris, setiap satu dengan sembilan aksara, di mana setiap aksara adalah sama ada `.` atau `*`, di mana `.` bermaksud sel tersebut kosong dan `*` bermaksud bintang diletakkan dalam sel tersebut.

- Sel kosong yang diberikan dijamin terdiri daripada aksara `1` hingga `9`.
- Sel kosong yang diberikan dijamin mempunyai tepat sembilan kawasan, setiap satu disambung empat arah. Maksudnya, anda sentiasa boleh bergerak dari satu sel ke sel lain dalam kawasan yang sama melalui sel jiran yang berkongsi tepi, dengan kesemuanya tergolong dalam kawasan yang sama.
- Penempatan bintang yang diberikan dijamin terdiri daripada aksara `.` dan `*`.

Output

Jika penempatan bintang memenuhi semua syarat, keluarkan `Valid`; jika ada syarat yang dilanggar, keluarkan `Invalid`.

Sample Input 1	Sample Output 1
<pre> 121333344 121334444 121111144 111111144 551116644 511176666 518777966 588877996 888887996 .*...*... ...*...*. .*...*... ...*...*. *...*... ...*...* ..*...*.. *...*... ..*...*.. *...*... ..*...*.. </pre>	<pre> Valid </pre>



Sample Input 2	Sample Output 2
<pre> 121333344 121334444 121111144 111111144 551116644 511176666 518777966 588877996 888887996 .*...*... ...*...*. .*.....* ...*.*. *.....*.. ...*.*. .*...*.. *...*... ..*...*. </pre>	Invalid

Sample Input 3	Sample Output 3
<pre> 121333344 121334444 121111144 111111144 551116644 511176666 518777966 588877996 888887996 .*..... ...*...*. .*...*... ...*...*. *.....*.. ...*.*. .*...*.. *...*... ..*...*. </pre>	Invalid



This page is intentionally left blank.

H. Membaiki Istana

Problem ID: repair



Little Po tinggal di sebuah istana di mana dia hidup bahagia setiap hari, sehingga suatu hari Little Yee menyerang istana tersebut. Dalam serangan itu, salah satu dinding istana dimusnahkan sepenuhnya! Secara kebetulan, Little Po berpendapat bahawa dinding itu sangat hodoh, jadi ini memberikan peluang yang baik untuk membinanya semula.

Lebar dinding itu adalah N meter, dan dia membahagikan dinding tersebut kepada N bahagian, masing-masing selebar 1 meter. Bahagian-bahagian ini dinomborkan dari 1 hingga N dari kiri ke kanan. Dia telah memesan Q batu bata, di mana setiap bata mempunyai lebar 1 meter, ketebalan yang sama dengan dinding, dan ketinggian yang berbeza-beza. Setiap bata mesti muat tepat ke dalam satu bahagian, tidak boleh merentasi lebih daripada satu bahagian, dan tidak boleh diputar.

Secara formal, setiap satu bahagian N ini mempunyai ketinggiannya sendiri. Pada mulanya, semua bahagian mempunyai ketinggian 0. Jika ketinggian asal sesuatu bahagian ialah x dan Little Po menambah bata dengan ketinggian y pada bahagian tersebut, maka ketinggian baru bahagian itu ialah $x + y$.

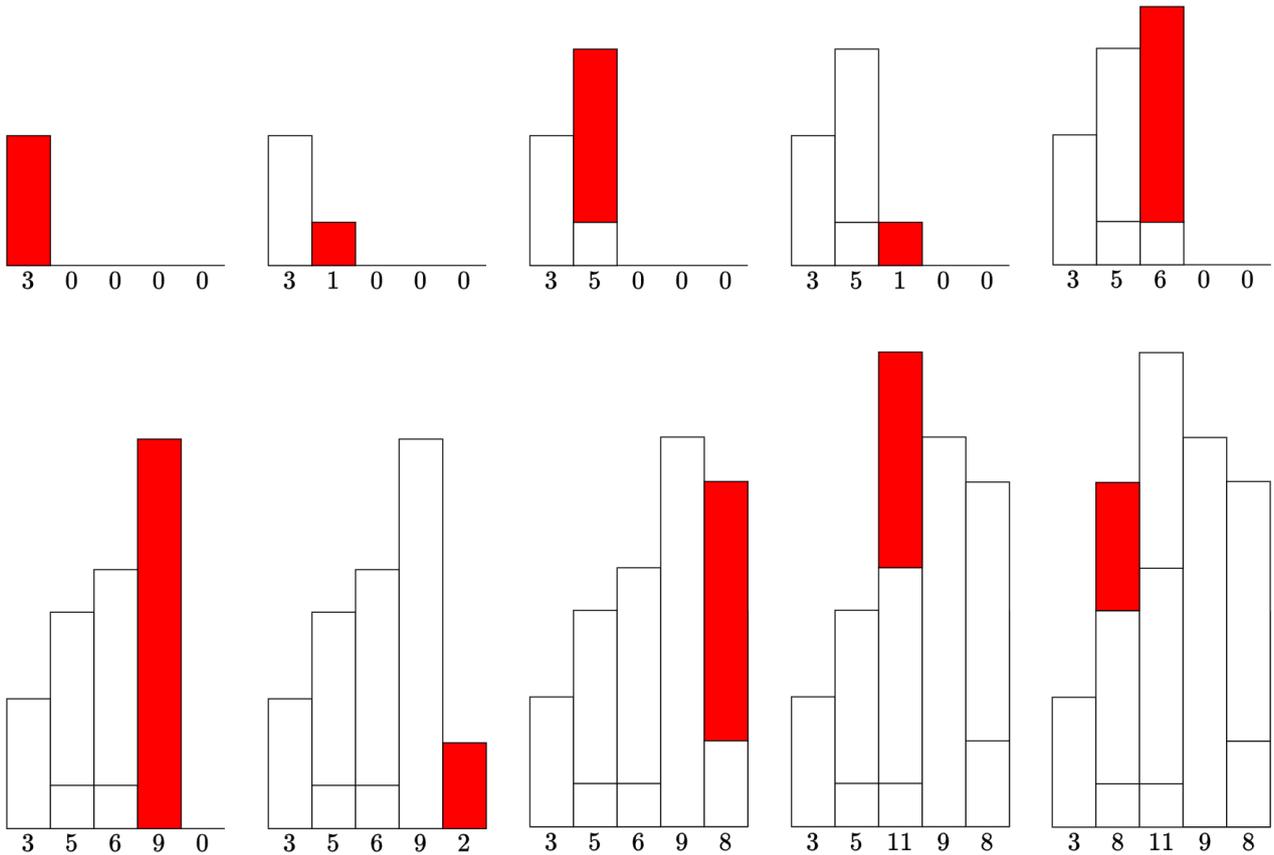
Batu bata yang ditempah, sebanyak Q bata, akan diterima satu persatu mengikut turutan tempahan. Ketinggian bata ke- i ialah h_i , tetapi Little Po tidak mengetahui ketinggian setiap bata sehingga ia sampai. Dia tidak sabar untuk melihat dinding itu disiapkan, jadi dia mesti meletakkan setiap bata sebaik sahaja ia sampai. Prinsip pemilihan bahagian adalah seperti berikut: Jika ketinggian semasa bahagian ke- i ialah x_i , **kehodohan** dinding ditakrifkan sebagai:

$$\sum_{k=0}^N (x_k - x_{k+1})^2$$

dengan syarat $x_0 = x_{N+1} = 0$, Little Po ingin memilih bahagian untuk meletakkan bata supaya kehodohan diminimumkan. Jika terdapat beberapa bahagian yang menghasilkan kehodohan minimum yang sama, dia akan memilih bahagian yang bernombor paling kecil.

Sebagai contoh, jika $N = 5$, $Q = 10$, dan ketinggian 10 bata adalah 3, 1, 4, 1, 5, 9, 2, 6, 5, 3 mengikut turutan. Apabila meletakkan bata pertama, tidak kira di mana ia diletakkan, kehodohan akan menjadi 18, jadi Little Po akan meletakkan bata pertama di bahagian pertama. Untuk bata kedua, jika ia diletakkan di bahagian 1, kehodohan akan menjadi 32; jika diletakkan di bahagian 2, kehodohan akan menjadi 14; dan jika diletakkan di bahagian lain, kehodohan

akan menjadi 20. Oleh itu, Little Po akan meletakkan bata kedua di bahagian 2. Perubahan dalam ketinggian setiap bahagian semasa proses ini ditunjukkan dalam rajah di bawah, dengan bata merah mewakili bata yang baru diletakkan, dan nombor di bawah menunjukkan ketinggian setiap bahagian.



Little Yee secara diam-diam mengetahui ketinggian kesemua Q batu bata tersebut dan ingin mengetahui bagaimana rupa dinding baru yang dibina oleh Little Po agar dia dapat merancang serangan seterusnya. Sila berikan Little Yee ketinggian akhir setiap bahagian dinding.

Input

Baris pertama mengandungi dua integer, N dan Q , yang mewakili lebar dinding dan bilangan bata, masing-masing.

Baris kedua mengandungi Q integer h_1, h_2, \dots, h_Q , yang mewakili ketinggian setiap bata dalam urutan ketibaan mereka.



- $1 \leq N, Q \leq 500$
- $1 \leq h_i \leq 10^5$

Output

Output N integer, x_1, x_2, \dots, x_N , di mana x_i mewakili ketinggian akhir setiap bahagian ke- i .

Sample Input 1

5 10 3 1 4 1 5 9 2 6 5 3	3 8 11 9 8
-----------------------------	------------

Sample Output 1

Sample Input 2

3 4 100000 100000 100000 100000	100000 200000 100000
------------------------------------	----------------------

Sample Output 2



This page is intentionally left blank.

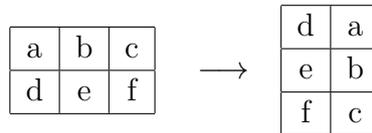
I. Operasi Jadual

Problem ID: operations



Anda diberikan jadual $N \times M$, dengan setiap sel mengandungi huruf kecil. Kita menggunakan (i, j) untuk mewakili sel dalam baris ke- i ($1 \leq i \leq N$) dari atas dan lajur ke- j ($1 \leq j \leq M$) dari kiri. Anda perlu melaksanakan Q operasi ke atas jadual secara berturutan. Terdapat dua jenis utama operasi.

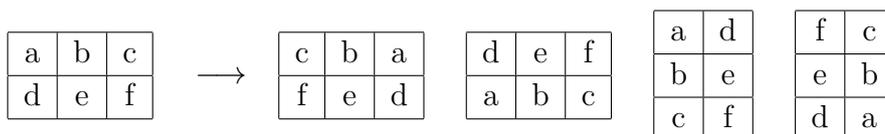
Jenis pertama adalah “putaran”. Setiap operasi putaran disertai dengan integer positif t , yang menunjukkan jadual perlu diputar 90 darjah t kali berturut-turut mengikut arah jam. Selepas satu putaran 90 darjah, jadual $R \times C$ akan menjadi jadual $C \times R$. Sel yang asalnya berada di (i, j) akan bergerak ke $(j, R + 1 - i)$. Sebagai contoh, jadual 2×3 di sebelah kiri akan menjadi jadual 3×2 di sebelah kanan selepas memutar 90 darjah sekali.



Jenis kedua adalah “flip”, yang boleh dibahagikan kepada empat jenis flip seperti berikut:

- Flip mendatar: Setelah flip mendatar, jadual $R \times C$ kekal sebagai $R \times C$, tetapi sel yang asalnya berada di (i, j) akan bergerak ke $(i, C + 1 - j)$.
- Flip menegak: Setelah flip menegak, jadual $R \times C$ kekal sebagai $R \times C$, tetapi sel yang asalnya berada di (i, j) akan bergerak ke $(R + 1 - i, j)$.
- Flip diagonal utama: Setelah flip diagonal utama, jadual $R \times C$ akan menjadi $C \times R$, dan sel yang asalnya berada di (i, j) akan bergerak ke (j, i) .
- Flip anti-diagonal: Setelah flip anti-diagonal, jadual $R \times C$ akan menjadi $C \times R$, dan sel yang asalnya berada di (i, j) akan bergerak ke $(C + 1 - j, R + 1 - i)$.

Sebagai contoh, jadual 2×3 di sebelah kiri akan menjadi empat jadual di sebelah kanan selepas flip mendatar, menegak, diagonal utama, dan anti-diagonal, masing-masing.



Selepas melaksanakan semua operasi, sila berikan rupa akhir jadual.



Input

Baris pertama: tiga integer N , M , dan Q .

Seterusnya, terdapat N baris, di mana baris ke- i mengandungi string s_i sepanjang M . Watak ke- j dalam s_i mewakili huruf kecil di sel (i, j) .

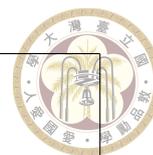
Seterusnya, terdapat Q baris, di mana baris ke- i mengandungi watak opi dan integer t_i dengan maksud berikut:

- Jika $op_i = \mathbf{R}$, maka operasi ke- i adalah operasi putaran. Jadual perlu diputar 90 darjah ti kali berturut-turut.
- Jika $op_i = \mathbf{F}$, maka operasi ke- i adalah operasi flip, di mana $t_i = 1, 2, 3, 4$ masing-masing mewakili flip mendatar, flip menegak, flip diagonal utama, dan flip anti-diagonal.
- $1 \leq N, M, Q \leq 100$
- Semua s_i terdiri daripada huruf kecil sahaja
- op_i adalah sama ada \mathbf{R} atau \mathbf{F}
- Jika $op_i = \mathbf{R}$, maka $1 \leq t_i \leq 10^9$.
- Jika $op_i = \mathbf{F}$, maka $1 \leq t_i \leq 4$.

Output

Pada baris pertama, keluarkan dua integer positif N' , M' yang menunjukkan bahawa selepas semua operasi, jadual yang dihasilkan mempunyai saiz $N' \times M'$.

Seterusnya, keluarkan N' baris, di mana baris ke- i mengandungi string s'_i sepanjang M' . Watak ke- j dalam s'_i mewakili huruf kecil di sel (i, j) dalam jadual akhir.

**Sample Input 1**

```
3 10 4
balutesih
littlecube
hsinmutsai
F 3
R 2
F 1
R 103
```

Sample Output 1

```
3 10
iastumnish
ebuceltil
hihsetulab
```

Sample Input 2

```
3 6 4
wiwiho
howiwi
wihowi
F 4
F 2
R 49
R 1
```

Sample Output 2

```
6 3
oii
hww
iio
wwh
ioi
whw
```



This page is intentionally left blank.

J. Pokok yang Membesar

Problem ID: tree



Old MacDonald mempunyai sebidang ladang yang boleh diwakili sebagai grid $N \times M$. Pada mulanya, sebuah pokok dengan ketinggian 0 ditanam di setiap posisi dalam grid tersebut. Kita menomborkan baris dari atas ke bawah dan lajur dari kiri ke kanan, dan menyatakan dengan (i, j) sel dalam baris ke- i dan lajur ke- j . Old MacDonald mempunyai banyak ejen pertumbuhan yang boleh digunakan pada sel (i, j) dengan syarat $2 \leq i \leq N - 1$ dan $2 \leq j \leq M - 1$. Selepas menggunakan ejen, ketinggian pokok di semua sel (x, y) yang memenuhi syarat $|x - i| \leq 1$ dan $|y - j| \leq 1$ akan meningkat sebanyak 1.

Pagi ini, Old MacDonald telah menggunakan beberapa ejen pertumbuhan. Malangnya, pada waktu petang, dia terlupa di mana dia telah menggunakannya. Kini, Old MacDonald memberitahu anda ketinggian pokok di setiap sel dan ingin anda membantu menentukan bilangan kali dia telah menggunakan ejen pertumbuhan di setiap kedudukan.

Input

Baris pertama input mengandungi dua integer positif N dan M , yang mewakili saiz grid.

Baris-baris berikutnya sebanyak N mengandungi M nombor setiap satu. Baris ke- i mengandungi $a_{i,1}, a_{i,2}, \dots, a_{i,M}$, di mana $a_{i,j}$ mewakili ketinggian pokok di sel (i, j) .

- $3 \leq N \leq 500$
- $3 \leq M \leq 500$
- $0 \leq a_{i,j} \leq 10^6$
- Dijamin bahawa Old MacDonald tidak tersilap mengira ketinggian pokok, dan ejen pertumbuhan adalah satu-satunya cara untuk meningkatkan ketinggian pokok.

Output

Keluarkan $N - 2$ baris. Baris ke- i mengandungi $M - 2$ integer $b_{i,1}, b_{i,2}, \dots, b_{i,M-2}$, di mana $b_{i,j}$ mewakili bilangan kali Old MacDonald menggunakan ejen pertumbuhan di sel $(i + 1, j + 1)$.



Sample Input 1

Sample Output 1

4 5 0 1 3 3 2 1 2 5 4 3 1 2 5 4 3 1 1 2 1 1	0 1 2 1 0 1
---	----------------

Sample Input 2

Sample Output 2

4 3 0 0 0 0 0 0 0 0 0 0 0 0	0 0
---	--------

K. Kiub Rubik

Problem ID: rubiks-cube



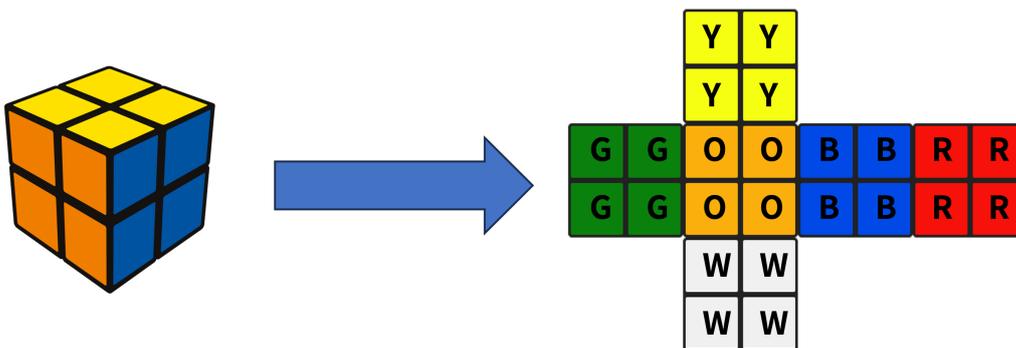
FHvirus sentiasa terpesona dengan kiub Rubik. Untuk mahir dalam menyelesaikan pelbagai jenis kiub Rubik, seseorang bukan sahaja perlu mampu menyelesaikannya, tetapi juga melakukannya dengan pantas. Oleh itu, FHvirus bukan sahaja meneroka pelbagai jenis kiub Rubik, malah turut mengambil bahagian dalam pelbagai pertandingan Kiub Rubik.

Sebagai penggemar kiub Rubik, FHvirus baru-baru ini mula belajar pengaturcaraan. Menggabungkan minatnya terhadap kiub Rubik dengan minat baru dalam pengaturcaraan, dia memutuskan untuk menulis “pencacah Kiub Rubik” bagi membantunya berlatih.

Apakah itu pencacah Kiub Rubik? Pemain kiub Rubik perlu mencacah dan menyelesaikan kiub berulang kali dengan pantas untuk berlatih. Jika kiub dicacah secara rawak setiap kali, kebiasaan manusia pasti akan membawa kepada corak pencacahan yang serupa. Oleh itu, menulis “pencacah” yang adil adalah sangat penting.

Oleh itu, “pencacah Kiub Rubik” menghasilkan urutan rawak “algoritma pencacahan” yang boleh diikuti oleh pemain untuk mencacah kiub. Selepas mencacah, pemain akan berlatih menyelesaikannya dengan cepat seperti biasa.

Namun, FHvirus baru sahaja mula belajar pengaturcaraan, jadi dia memutuskan untuk bermula dengan “pencacah Kiub Rubik 2x2” yang paling mudah. Pertama, kita perlu memahami apa itu “kiub Rubik 2x2”, seperti yang ditunjukkan dalam gambar di bawah.



Dalam gambar di atas, bahagian kiri menunjukkan kiub Rubik 2x2, yang mempunyai enam permukaan, setiap satu terdiri daripada empat petak, biasanya berwarna kuning, oren, biru, merah, hijau, dan putih. Bahagian kanan adalah diagram yang telah dibuka, menunjukkan warna-warna dari keenam-enam permukaan, dengan kuning (Yellow), oren (Orange), biru (Blue),

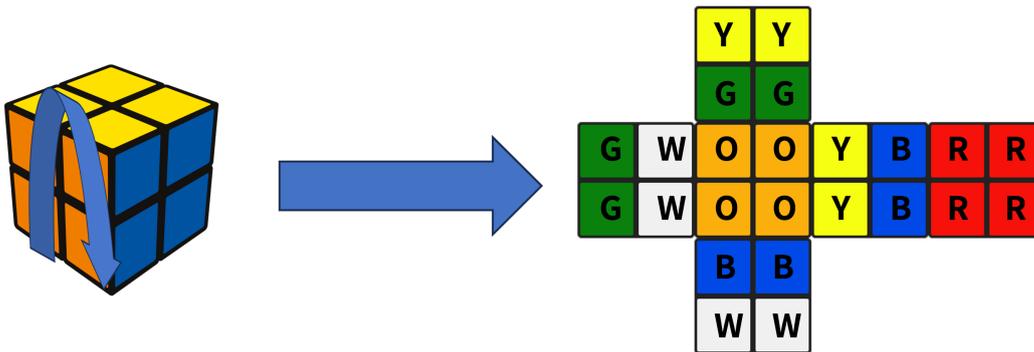
merah (Red), hijau (Green), dan putih (White) masing-masing disingkat sebagai **Y**, **O**, **B**, **R**, **G**, dan **W**.



Seterusnya, kami memperkenalkan tiga operasi asas pada kiub Rubik 2x2.

1. **F** Operasi

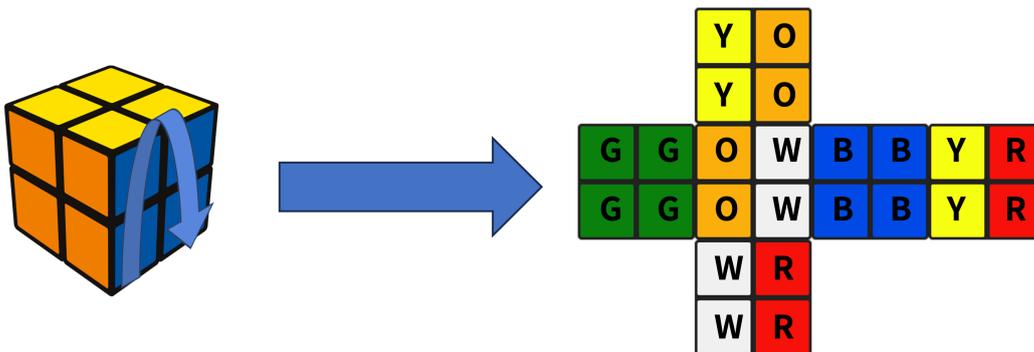
Operasi **F** memutar muka hadapan Rubik’s Cube mengikut arah jam, seperti yang ditunjukkan di bawah:



Dalam gambar di atas, bahagian kiri menunjukkan muka yang berputar dan arah putaran, manakala bahagian kanan menunjukkan gambar yang terbentang selepas putaran. Untuk memudahkan penulisan, **F2** menunjukkan “memutar muka hadapan Rubik’s Cube sebanyak 180 darjah,” yang bersamaan dengan melaksanakan operasi **F** dua kali; **F'** menunjukkan “memutar muka hadapan Rubik’s Cube mengikut arah lawan jam,” yang bersamaan dengan melaksanakan operasi **F** tiga kali.

2. **R** Operasi

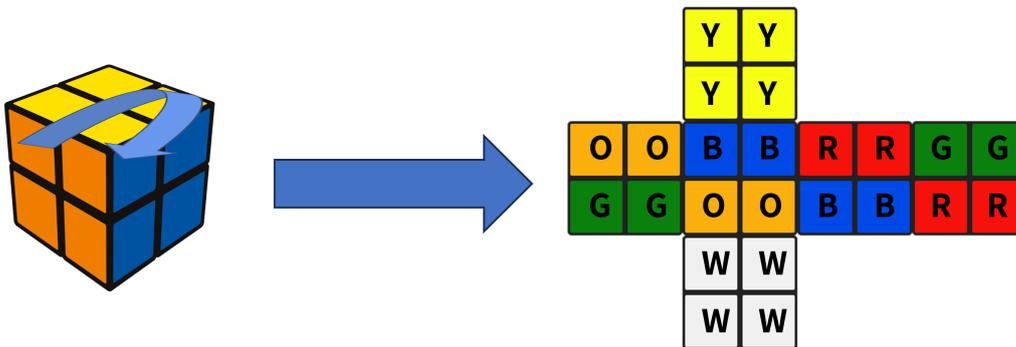
Operasi **R** memutar muka kanan Rubik’s Cube mengikut arah jam, seperti yang ditunjukkan di bawah:



Dalam gambar di atas, sebelah kiri menunjukkan muka yang diputar dan arah putaran, manakala sebelah kanan menunjukkan diagram yang telah dikembangkan selepas putaran. Untuk mempermudah notasi, **R2** bermaksud “memutar muka kanan Rubik’s Cube sebanyak 180 darjah,” yang sama dengan melakukan operasi **R** dua kali; **R'** bermaksud “memutar muka kanan Rubik’s Cube mengikut arah lawan jam,” yang sama dengan melakukan operasi **R** tiga kali.

3. U Operasi

Operasi **U** memutar muka atas Kiub Rubik mengikut arah jam, seperti yang ditunjukkan di bawah:



Dalam imej di atas, sebelah kiri menunjukkan muka dan arah yang berputar, dan sebelah kanan menunjukkan gambar rajah yang tidak dilipat selepas putaran. Untuk memudahkan tatatanda, **U2** menunjukkan “memusingkan muka atas Kiub Rubik 180 darjah,” yang bersamaan dengan melakukan operasi **U** dua kali; **U'** menunjukkan “memusingkan muka atas Kiub Rubik melawan arah jam,” yang bersamaan dengan melakukan operasi **U** tiga kali.

Dengan operasi asas ini di tangan, tiba masanya untuk menulis 2x2 Rubik’s Cube scrambler. Pertama, kita perlu menulis “penjana algoritma berebut” untuk menjana urutan operasi asas. Kemudian, kita perlu menulis “simulator putaran” untuk mensimulasikan keadaan Kiub Rubik 2x2 selepas mengikuti algoritma perebutan, supaya pemain boleh mengesahkan sama ada mereka berebut dengan betul.

FHVirus telah pun menulis bahagian penjana, tetapi dia menghadapi kesukaran dengan bahagian simulator. Bolehkah anda membantunya menulis simulator yang membaca algoritma perebutan dan mengeluarkan keadaan Kiub Rubik 2x2 selepas mengikuti algoritma perebutan?



Input

Baris pertama input mengandung integer positif N , mewakili panjang algoritma perebutan yang dihasilkan.

Baris seterusnya mengandung rentetan N , setiap satu mewakili satu daripada sembilan operasi asas: $F, F', F2, R, R', R2, U, U'$ dan $U2$.

- $1 \leq N \leq 100$

Output

Keluarkan gambarajah Rubik's Cube yang tidak dilipat selepas mengikuti algoritma perebutan, bermula dari keadaan awal yang ditunjukkan dalam imej pertama dalam pernyataan masalah. Untuk gambar rajah yang tidak dilipat, sistem penghakiman akan melakukan perbandingan yang ketat, bermakna anda mesti mengeluarkan mengikut format berikut:

```
YY
YY
GGOOBBRR
GGOOBBRR
WW
WW
```

Baris pertama, kedua, kelima dan keenam masing-masing mempunyai dua aksara ruang utama (kod ASCII 32), dan setiap baris berakhir dengan aksara baris baharu (kod ASCII 10). Ambil perhatian bahawa baris pertama, kedua, kelima dan keenam tidak mempunyai ruang mengekor tambahan, menjadikan panjangnya lebih pendek sebanyak empat aksara berbanding baris ketiga dan keempat.

Sample Input 1

```
8
U F R F R U F R
```

Sample Output 1

```
WB
RG
OBWWORDYB
GYGOYGYR
RB
WO
```

Sample Input 2

Sample Output 2



<p>10 F2 R' U F' R2 U2 F R' F' R</p>	<p>OY RG GBWYROBY GRBBWOWR YO WG</p>
--	--



This page is intentionally left blank.

L. Pembuat Obor

Problem ID: torch



LittleCube suka bermain permainan kiub dan baru-baru ini telah menemui permainan kiub baharu. Dalam permainan ini, terdapat lima jenis barang: kayu balak, kayu, arang, batang kayu, dan obor. LittleCube pada masa ini mempunyai N kayu balak dan dia sangat menyukai obor, jadi dia ingin membuat sebanyak mungkin obor menggunakan kayu balak ini. Bagaimanakah LittleCube boleh membuat obor? Dia boleh melakukan dua jenis operasi: **pembuatan** dan **peleburan**.

- **Pembuatan:** Dengan menggunakan bahan yang diperlukan mengikut resipi, dia boleh mendapatkan item yang ditentukan. Berikut adalah semua resipi pembuatan, di mana mengambil item di sebelah kiri anak panah akan menghasilkan item di sebelah kanan.

1 kayu balak \rightarrow 4 kayu

2 kayu \rightarrow 4 batang kayu

1 batang kayu dan 1 arang \rightarrow 1 obor

- **Peleburan:** Satu operasi peleburan boleh menukar 1 kayu balak kepada 1 arang, tetapi ia memerlukan 1 unit tenaga. Tenaga boleh diperolehi dengan menggunakan bahan api. Semua barang kecuali obor boleh digunakan sebagai bahan api. Jadual di bawah menunjukkan berapa banyak unit tenaga yang diperolehi dari setiap jenis item apabila digunakan sebagai bahan api.

Bahan api	Tenaga
1 kayu balak	1 unit
1 kayu	1 unit
1 batang kayu	0.2 unit
1 arang	8 unit

Ambil perhatian bahawa untuk menggunakan item sebagai bahan api, keseluruhan item mesti digunakan. Sebagai contoh, anda tidak boleh menggunakan hanya $1/8$ daripada arang untuk mendapatkan 1 unit tenaga.

Sebagai contoh, jika LittleCube bermula dengan 10 kayu balak, dia boleh menggunakan 3 kayu balak untuk membuat $3 \times 4 = 12$ kayu. Dengan menggunakan 7 daripada kayu ini sebagai bahan api, dia akan memperolehi 7 unit tenaga, yang boleh digunakan untuk menukarkan baki 7 kayu balak menjadi arang. Dia kemudian boleh menggunakan 4 daripada 5 kayu yang tinggal

untuk membuat $\frac{4}{2} \times 4 = 8$ batang kayu. Akhirnya, dia boleh menggunakan 7 batang kayu dan 7 arang untuk membuat 7 obor.



Sila beritahu LittleCube bilangan maksimum obor yang boleh dibuatnya.

Input

Baris pertama mengandungi integer T , yang mewakili bilangan kes ujian.

Setiap satu daripada baris berikutnya sebanyak T baris mengandungi integer tunggal N , yang mewakili bilangan kayu balak yang dimiliki oleh LittleCube.

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^{12}$

Output

Outputkan T baris, di mana setiap baris ke- i mengandungi satu integer yang mewakili jawapan untuk kes ujian ke- i .

Sample Input 1	Sample Output 1
14	0
1	1
2	2
3	2
4	3
5	4
6	4
7	5
8	6
9	7
10	78
100	214399693511
271828182845	247787589578
314159265359	788732394365
1000000000000	

M. Data Eksperimen

Problem ID: experiments



Charlie telah menjalankan beberapa penyelidikan baru-baru ini. Penyelidikan ini melibatkan pelaksanaan beberapa eksperimen, di mana setiap eksperimen menghasilkan markah yang mewakili keputusannya. Charlie telah menjalankan sejumlah N eksperimen, di mana markah eksperimen ke- i ialah a_i .

Charlie perlu menganalisis data yang diperolehinya. Dalam proses analisis, dia perlu mengira varians untuk beberapa selang tertentu. Lebih khusus lagi, terdapat Q pasangan selang (l_i, r_i) . Untuk setiap pasangan (l_i, r_i) , Charlie perlu mengira varians bagi $a_{l_i}, a_{l_i+1}, \dots, a_{r_i}$.

Charlie tidak tahu cara untuk mengira varians Q ini dengan cepat, jadi dia meminta bantuan anda. Sila bantu Charlie dalam mengira varians Q yang dia perlukan.

Varians bagi K nombor nyata x_1, x_2, \dots, x_K ditakrifkan sebagai $\frac{1}{K} \sum_{i=1}^K (x_i - \bar{x})^2$, di mana $\bar{x} = \frac{1}{K} \sum_{i=1}^K x_i$.

Input

Baris pertama mengandungi dua integer N dan Q .

Baris kedua mengandungi N integer a_1, a_2, \dots, a_N .

Setiap baris daripada Q baris berikutnya mengandungi dua integer l_i dan r_i .

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $-10^6 \leq a_i \leq 10^6$
- $1 \leq l_i \leq r_i \leq N$

Output

Keluarkan Q baris. Baris ke- i harus mengandungi integer yang mewakili $(r_i - l_i + 1)^2 \cdot V_i$, di mana V_i ialah varians bagi $a_{l_i}, a_{l_i+1}, \dots, a_{r_i}$.

Ia boleh dibuktikan bahawa $(r_i - l_i + 1)^2 \cdot V_i$ sentiasa merupakan integer di bawah syarat-syarat yang diberikan.

**Sample Input 1**

5 3	24
1 5 3 2 4	0
1 3	50
4 4	
1 5	

Sample Output 1**Sample Input 2**

6 3	177624
-49 49 49 -49 100 -100	0
1 6	64802
2 3	
4 6	

Sample Output 2

N. Kemaskan Desktop

Problem ID: desktop



Pernahkah anda meletakkan terlalu banyak fail secara tidak sengaja di desktop komputer anda, menjadikannya bersepah sangat? Orang yang biasa memastikan persekitaran mereka kemas mungkin tidak menghadapi masalah ini, tetapi 8e7 bukanlah salah seorang daripada golongan tersebut. Pada suatu hari, 8e7 menyedari bahawa desktopnya bersepah sangat, terutamanya disebabkan oleh dia telah meninggalkan terlalu banyak fail yang tidak diperlukan di sana sebagai simpanan sementara. Dengan memadamkan fail-fail ini, desktopnya boleh menjadi jauh lebih kemas.

Walau bagaimanapun, keengganan 8e7 untuk memastikan persekitarannya kemas adalah akibat daripada sifat malasnya. Oleh itu, dia mahu mengeluarkan usaha yang paling sedikit untuk memadamkan fail-fail ini. 8e7 mendapati bahawa dengan memilih satu segi empat di desktop, dia boleh menukar status fail-fail dalam segi empat itu kepada status pemilihan. Mengklik padam kemudiannya akan memadamkan semua fail yang dipilih sekaligus. Tetapi apabila memilih segi empat kedua, dia menyedari bahawa perkara itu tidak semudah yang disangkakan!

Sebenarnya, sistem operasi komputer 8e7 direka sedemikian rupa sehingga setiap kali dia memilih satu segi empat di desktop, status pemilihan semua fail dalam segi empat itu akan ditukar.

Apakah maksud ini? Ini bermakna bahawa mana-mana fail dalam segi empat itu yang sebelum ini tidak dipilih akan dipilih, dan mana-mana fail yang sudah dipilih tidak akan dipilih!

Operasi ini menjadi agak rumit, tetapi 8e7 masih mahu mengeluarkan usaha yang paling sedikit untuk memilih semua fail supaya dia boleh memadamkan semua fail dengan satu klik. Sekarang, 8e7 telah menukar semua fail kepada N koordinat pada satah 2D, di mana M fail daripada fail-fail ini perlu dipadamkan. Sila tulis satu program yang, selepas membaca koordinat semua fail, menyediakan 8e7 dengan cara untuk memilih semua fail menggunakan jumlah **operasi pemilihan segi empat yang paling minimum**.

Input

Baris pertama input mengandungi dua integer positif N dan M , yang mewakili jumlah bilangan fail di desktop 8e7 dan bilangan fail yang perlu dipadamkan.

Dalam N baris berikutnya, baris ke- i mengandungi dua integer x_i dan y_i , yang mewakili

koordinat fail ke- i . Fail bernombor 1 hingga M ialah fail-fail yang ingin dipadamkan oleh 8e7



- $1 \leq M \leq N \leq 14$
- $-10^9 \leq x_i, y_i \leq 10^9$
- Dijamin bahawa semua koordinat fail adalah berbeza.

Output

Baris pertama output harus mengandungi satu integer positif k , yang mewakili bilangan operasi pemilihan segi empat yang diperlukan.

Dalam k baris berikutnya, baris ke- j harus mengandungi empat integer l_j, d_j, r_j, u_j , yang mewakili koordinat segi empat yang dibentuk oleh (l_j, d_j) dan (r_j, u_j) untuk operasi pemilihan ke- j . Segi empat ini mesti memenuhi syarat $-2 \times 10^9 \leq l_j \leq r_j \leq 2 \times 10^9$ dan $-2 \times 10^9 \leq d_j \leq u_j \leq 2 \times 10^9$. Untuk fail ke- i , jika $l_j \leq x_i \leq r_j$ dan $d_j \leq y_i \leq u_j$, status pemilihannya akan ditukar.

Cara pemilihan output anda mesti memastikan bahawa fail bernombor 1 hingga M berada dalam keadaan terpilih, manakala fail lain kekal tidak dipilih. Jika terdapat pelbagai penyelesaian yang mungkin, keluarkan mana-mana satu daripadanya kerana semuanya akan dianggap betul.

Sample Input 1

```
5 3
1 4
2 2
5 4
3 3
4 5
```

Sample Output 1

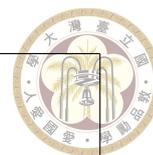
```
2
1 2 5 5
3 3 4 5
```

Sample Input 2

```
6 3
-3 5
-3 9
6 0
6 5
-3 8
6 2
```

Sample Output 2

```
3
-3 0 6 0
-3 9 -3 9
-3 0 -3 5
```





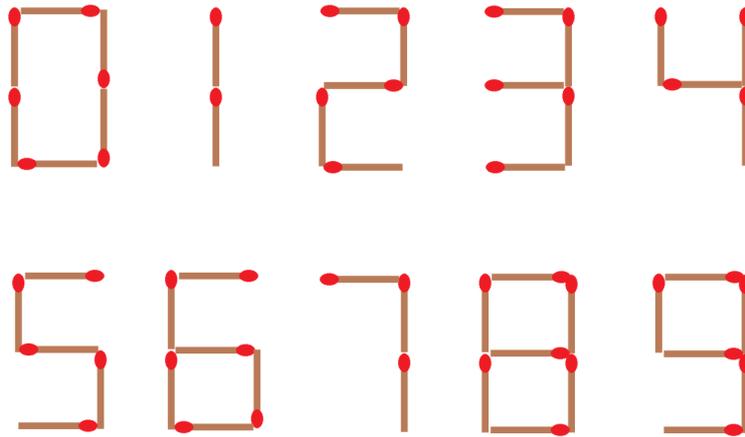
This page is intentionally left blank.

O. Batang Mancis Ai-chan

Problem ID: matchsticks



Ia diketahui umum bahawa batang mancis boleh digunakan untuk membuat nombor. Cara untuk membuat digit 0 hingga 9 dengan batang mancis ditunjukkan di gambar berikut:



Integer yang berbeza memerlukan nombor batang mancis yang tak sama. Contohnya, untuk membuat nombor 17, kita perlu digit 1 dan 7, jadi memerlukan jumlah $2 + 3 = 5$ batang mancis. Untuk membuat nombor 45510, kita memerlukan $4 + 5 + 5 + 2 + 6 = 22$ batang mancis.

Ai-chan ada N batang mancis dan mahu guna semuanya untuk membuat satu integer bukan negatif.

Berapa nombor bukan negatif yang boleh Ai-chan buat? Disebabkan jawapan boleh jadi sangat besar, sila output baki jawapan selepas dibahagikan dengan 998244353.

Input

Input mengandungi satu integer positif N .

- $1 \leq N \leq 10^{18}$

Output



Sila keluarkan satu integer bukan negatif yang mewakili bilangan integer bukan negatif yang berbeza Ai-chan boleh buat, output baki jawapan selepas dibahagikan dengan 998244353.

Sample Input 1

7

Sample Output 1

12

Sample Input 2

3

Sample Output 2

1

Sample Input 3

1000000

Sample Output 3

811496257
