Question: Student Marks Verification

You are a teacher who maintains an **official marks list** for students. An assistant also records marks during exams. At the end, you want to verify if the assistant recorded marks correctly.

Write a function countWrongEntries() that returns the number of mismatches between the official list and assistant's list.

Function Parameters:

- students: an array of n strings (student names).
- marks: an array of n integers (official marks).
- recorded_names: an array of m strings (names in assistant's record).
- recorded_marks: an array of m integers (marks in assistant's record).

Input Format

n

student1

student2

...

studentN

mark1

mark2

...

markN

m

recorded_student1

recorded_student2

...

recorded_studentM

recorded_mark1

recorded_mark2

...

recorded_markM

Output Format

Print the number of mismatches in marks between the two lists.

Example Input

4

Amit

Riya

Kunal

Meera

85

92

76

88

3

Riya

Kunal

Meera

90

76

88

Example Output

1

Explanation

- Official list: {Amit=85, Riya=92, Kunal=76, Meera=88}
- Assistant list: {Riya=90, Kunal=76, Meera=88}
- Mismatch → Riya (92 vs 90).
  So, answer = 1.

Concepts Involved

1. Build a HashMap<String, Integer> from official list.

2. Iterate over assistant’s list, lookup in map.
3. Compare values → count mismatches.
4. Return mismatch count.

Question: Library Book Verification

You are the librarian of a public library. You maintain an **official catalog** of books along with their prices. An assistant is responsible for recording the daily sales of books along with the price at which they were sold. At the end of the day, you want to verify if the assistant has recorded the prices correctly.

Your task is to complete the function countWrongEntries() that returns the number of mismatches between the **official catalog** and the **sales record** of the assistant.

Function Parameters

The function takes the following arguments:

- books: an array of n strings, representing the names of books in the official catalog.
- prices: an array of n floating numbers, representing the official prices of the books.
- sold_books: an array of m strings, representing the names of books sold (as recorded by the assistant).
- sold_prices: an array of m floating numbers, representing the prices of the sold books (as recorded by the assistant).

Input Format

- The first line contains an integer n, the number of books in the catalog.
- The next n lines each contain a string (book name).
- The next n lines each contain a floating-point number (price of that book).
- The next line contains an integer m, the number of sold books in the assistant's record.
- The next m lines each contain a string (sold book name).
- The next m lines each contain a floating-point number (sold price of the book).

Output Format

- Print a single integer, the number of mismatches between the official catalog and the assistant's sales record.

Sample Input

5

HarryPotter

Narnia

Sherlock

Inferno

Hamlet

320.50

280.00

250.75

400.00

150.25

3

Narnia

Hamlet

Inferno

280.00

160.25

410.00

Sample Output

2

Explanation

- Official catalog:
- HarryPotter → 320.50
- Narnia → 280.00
- Sherlock → 250.75
- Inferno → 400.00
- Hamlet → 150.25
- Assistant record:
- Narnia → 280.00 (✅ correct)

- Hamlet → 160.25 (❌ mismatch, should be 150.25)
- Inferno → 410.00 (❌ mismatch, should be 400.00)
- Total mismatches = **2**.

Question: Employee Salary Cross-Check

You are the owner of a company and you maintain an **official record** of all employee salaries. The HR assistant submits a **payroll report** containing employee names and their reported salaries. At the end of the month, you want to verify if the HR assistant has reported the salaries correctly.

Your task is to complete the function countSalaryMismatch() that returns the number of mismatches between the **official salary list** and the **payroll report** submitted by the assistant.

Function Parameters

The function takes the following arguments:

- employees: an array of n strings, representing the names of employees in the official record.
- salaries: an array of n floating numbers, representing the official salaries of employees.
- reported_names: an array of m strings, representing the employee names in the payroll report.
- reported_salaries: an array of m floating numbers, representing the reported salaries of employees.

Input Format

- The first line contains an integer n, the number of employees in the official record.
- The next n lines each contain a string (employee name).
- The next n lines each contain a floating-point number (official salary of that employee).
- The next line contains an integer m, the number of employees in the payroll report.
- The next m lines each contain a string (employee name in the payroll report).
- The next m lines each contain a floating-point number (reported salary of that employee).

Output Format

- Print a single integer, the number of mismatches between the official salary list and the payroll report.

Sample Input

4

Amit

Riya

Kunal

Meera

50000.00

60000.00

55000.00

58000.00

3

Riya

Kunal

Meera

61000.00

55000.00

57000.00

Sample Output

2

Explanation

- Official record:
- Amit → 50000.00
- Riya → 60000.00
- Kunal → 55000.00
- Meera → 58000.00
- Payroll report:
- Riya → 61000.00 (❌ mismatch, should be 60000.00)
- Kunal → 55000.00 (✅ correct)
- Meera → 57000.00 (❌ mismatch, should be 58000.00)

So, the total mismatches = **2**.

Question 1: Exam Seating Verification

In an exam hall, students are seated according to a predefined seating plan (name → roll number). The invigilator records the seating order and roll numbers as observed during the exam.

As the exam coordinator, you need to verify if both the **roll numbers and their order** match exactly with the official seating plan.

Your task is to complete the function countSeatingMismatch() that returns the number of mismatches between the official seating plan and the invigilator's recorded seating.

Function Parameters

- students: an array of n strings, representing the names of students in the official plan.
- rolls: an array of n integers, representing their roll numbers.
- recorded_students: an array of m strings, representing student names in the recorded plan.
- recorded_rolls: an array of m integers, representing roll numbers in the recorded plan.

Input Format

- First line: integer n (number of students).
- Next n lines: student names.
- Next n lines: roll numbers.
- Next line: integer m (size of recorded plan).
- Next m lines: recorded student names.
- Next m lines: recorded roll numbers.

Output Format

- Print the number of mismatches in sequence or roll numbers.

Sample Input

4

Amit

Riya

Kunal

Meera

101

102

103

104

4

Amit

Riya

Meera

Kunal

101

102

104

103

Sample Output

2

Explanation

- Official order:
- Amit → 101
- Riya → 102
- Kunal → 103
- Meera → 104
- Recorded order:
- Amit → 101 (✅ correct)
- Riya → 102 (✅ correct)
- Meera → 104 (❌ wrong order, should be at last)
- Kunal → 103 (❌ wrong order, swapped)

So, mismatches = **2**.

Question 2: Playlist Order Verification

A music app maintains an official **playlist order** (song name → duration). A DJ creates their own playlist but claims it matches the official one. At the end of the event, you must check how many songs are **out of place or with wrong duration**.

Write a function countPlaylistMismatch() that returns the number of mismatches.

Input Format

- First line: integer n (number of songs).
- Next n lines: song names.
- Next n lines: durations (in minutes, floating).
- Next line: integer m (songs in DJ playlist).
- Next m lines: song names.
- Next m lines: durations.

Output Format

- Print the number of mismatches (either **wrong duration** or **wrong sequence**).

Sample Input

5

SongA

SongB

SongC

SongD

SongE

3.5

4.2

2.8

5.0

4.5

5

SongA

SongC

SongB

SongD

SongE

3.5

2.8

4.2

5.1

4.5

Sample Output

2

Explanation

- Official playlist:
- SongA → 3.5
- SongB → 4.2
- SongC → 2.8
- SongD → 5.0
- SongE → 4.5
- DJ playlist:
- SongA → 3.5 (✅ correct)
- SongC → 2.8 (❌ wrong order, should be after SongB)
- SongB → 4.2 (❌ wrong order, swapped with SongC)
- SongD → 5.1 (❌ wrong duration, should be 5.0)
- SongE → 4.5 (✅ correct)

So mismatches = **2** (order swap of B & C counts as 1, wrong duration of D counts as 1).

ArrayList

Question 1: Unique Visitors in a Website

You are given a list of visitors (names) who visited your website in a day. Some visitors visited multiple times.
Your task is to find the **number of unique visitors** using ArrayList.

Input Format

- First line: integer n (number of entries).
- Next n lines: strings (visitor names).

Output Format

- Print the number of unique visitors.

Example Input

6

Amit

Riya

Amit

Kunal

Meera

Riya

Example Output

3

Question 2: Reverse Playlist

A playlist of songs is stored in an ArrayList<String>. Write a program to **reverse the playlist** (without using Collections.reverse()).

Input Format

- First line: integer n (number of songs).
- Next n lines: strings (song names).

Output Format

- Print the reversed playlist (one per line).

Example Input

4

SongA

SongB

SongC

SongD

Example Output

SongD

SongC

SongB

SongA

Question 3: Find Common Students in Two Courses

Two ArrayList<String> contain student names enrolled in **Course A** and **Course B**.
Write a program to find the **list of students enrolled in both courses**.

Input Format

- First line: integer n1 (size of first list).
- Next n1 lines: student names.
- Next line: integer n2 (size of second list).
- Next n2 lines: student names.

Output Format

- Print the names of students present in both lists.

Example Input

4

Amit

Riya

Kunal

Meera

3

Kunal

Riya

Sita

Example Output

Riya

Kunal

Question 4: Remove Consecutive Duplicates

You are given an ArrayList<Integer> containing numbers. Write a program to **remove consecutive duplicates** (keep only one occurrence).

Input Format

- First line: integer n (size of list).
- Next n lines: integers.

Output Format

- Print the updated list after removing consecutive duplicates.

Example Input

8

1

1

2

2

2

3

1

1

Example Output

1 2 3 1