# Problem 1: Spiraled Message (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **SpiraledMessage.java**.

## Description

A spiraled message is a string that spirals inwards inside a rectangular grid. Given a string and a rectangular area of R rows and N columns, spiral the string around the inside of the rectangular grid (going inward) repeating the string whenever it finishes.

Here is an example with the string "EXAMPLE" and a grid of size 5 by 6.

```
EXAMPL
MPLEEE
AEEXXE
XLPMAX
EELPMA
```

For this version, you do not have to actually spiral the message. You only have to report the last character printed (in the inner most spot). In this case, it is the letter X, shown in bold and underlined here.

```
EXAMPL
MPLEEE
AEEXXE
XLPMAX
EELPMA
```

## Input

The first line contains the number N of cases with 0 < N <= 100.
This is then followed by N cases.
Each case consists of two integers on one line giving the number of rows R and number of columns C followed by the message string M on a second line. Each string can consist of the letters A-Z. No other characters will be used.

For boundaries, we will have 0 < R <= 100 and 0 < C <= 100 and 0 < length(M) <= 100.

## Output

The output consists of one line for each case, that is the last character printed in the spiral journey.

### Example Input

```
4
5 6
EXAMPLE
4 20
EXAMPLESENTENCE
1 8
THISISASHORTONE
3 3
ABC
```

### Example Output

```
X
P
S
C
```

# Problem 2: Spiraled Message Board (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **SpiraledMessageBoard.java**.

## Description

A spiraled message is a string that spirals inwards inside a rectangular grid. Given a string and a rectangular area of R rows and N columns, spiral the string around the inside of the rectangular grid (going inward) repeating the string whenever it finishes.

Here is an example with the string "EXAMPLE" and a grid of size 5 by 6.

```
EXAMPL
MPLEEE
AEEXXE
XLPMAX
EELPMA
```

***For this version, you will output the entire message.***

## Input

The first line contains the number N of cases with 0 < N <= 100.
This is then followed by N cases.
Each case consists of two integers on one line giving the number of rows R and number of columns C followed by the message string M on a second line. Each string can consist of the letters A-Z. No other characters will be used.

For boundaries, we will have 0 < R <= 100 and 0 < C <= 100 and 0 < length(M) <= 100.

## Output

The output consists of one spiraled message for each case, with no additional spacing.

### Example Input

```
4
5 6
EXAMPLE
4 20
EXAMPLESENTENCE
1 8
THISISASHORTONE
3 3
ABC
```

### Example Output

```
EXAMPL
MPLEEE
AEEXXE
XLPMAX
EELPMA
EXAMPLESENTENCEEXAMP
CEEXAMPLESENTENCEEXL
NPMAXEECNETNESELPMAE
ETNESELPMAXEECNETNES
THISISAS
ABC
BCA
ACB
```

# Problem 3: Pattern Matching (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **PatternMatch.java**.

## Description

Suppose you have a two-dimensional array A of bits (0 or 1 values) with RA rows and CA columns and a second array B of bits with RB rows and CB columns. Your task is to determine how many times the pattern B shows up inside A, with overlaps counting.

For example, suppose A is the following 6x6 array:

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

And suppose B is the following 3x3 array:

```
1 0 0
1 1 0
0 1 1
```

Then B shows up two times in A. Here is one occurrence (in bold):

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

And here is the other occurrence:

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

Note it must be an exact match to count. There is no wrapping around to the other side either.

## Input

The first line contains the number N of cases with 0 < N <= 100.
This is then followed by N cases.
Each case consists of two patterns. Each pattern starts with two integers on one line giving the number of rows R and number of columns C for that pattern followed by R lines each line containing C values (0 or 1).

For boundaries, we will have 0 < R <= 100 and 0 < C <= 100.

## Output

The output consists of one line for each case, that is the number of matches (the number of times the second pattern appears in the first pattern).

### Example Input

```
3
6 6
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
3 3
1 0 0
1 1 0
0 1 1
8 10
1 1 1 1 1 1 1 1 1 1
1 0 1 0 0 0 1 0 1 1
0 0 1 1 1 0 1 1 1 1
0 1 1 0 1 0 1 0 1 0
0 0 1 0 0 0 1 1 0 0
1 1 1 0 0 0 1 1 1 1
1 0 1 0 1 1 1 0 1 0
0 0 0 0 1 0 0 0 0 0
3 2
1 1
1 0
```

Suppose you have a two-dimensional array A of bits (0 or 1 values) with RA rows and CA columns and a second array B of bits with RB rows and CB columns. Your task is to determine how many times the pattern B shows up inside A, with overlaps counting.

For example, suppose A is the following 6x6 array:

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

And suppose B is the following 3x3 array:

```
1 0 0
1 1 0
0 1 1
```

Then B shows up two times in A. Here is one occurrence (in bold):

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

And here is the other occurrence:

```
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
```

Note it must be an exact match to count. There is no wrapping around to the other side either.

## Input

The first line contains the number N of cases with 0 < N <= 100.
This is then followed by N cases.
Each case consists of two patterns. Each pattern starts with two integers on one line giving the number of rows R and number of columns C for that pattern followed by R lines each line containing C values (0 or 1).

For boundaries, we will have 0 < R <= 100 and 0 < C <= 100.

## Output

The output consists of one line for each case, that is the number of matches (the number of times the second pattern appears in the first pattern).

### Example Input

```
3
6 6
0 0 1 1 0 1
1 1 0 0 1 1
0 1 1 0 0 1
0 0 1 1 0 0
1 1 0 1 1 1
0 1 0 1 1 1
3 3
1 0 0
1 1 0
0 1 1
8 10
1 1 1 1 1 1 1 1 1 1
1 0 1 0 0 0 1 0 1 1
0 0 1 1 1 0 1 1 1 1
0 1 1 0 1 0 1 0 1 0
0 0 1 0 0 0 1 1 0 0
1 1 1 0 0 0 1 1 1 1
1 0 1 0 1 1 1 0 1 0
0 0 0 0 1 0 0 0 0 0
3 2
1 1
1 0
0 0
10 4
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
1 1 0 1
1 1 1 1
1 1 1 1
1 1 1 1
2 2
1 1
1 1
```

### Example Output

```
2
5
23
```

# Problem 4: Anagrams (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **Anagram.java**.

## Description

Two strings are anagrams if you can shuffle the letters in one string and get the other string. For example, DORMITORY and DIRTYROOM are anagrams. Given a list of strings, count how many pairs of strings are anagrams of each other.

## Input

The first line contains the number N of cases with 0 < N <= 50.
This is then followed by N cases. Each case starts with a number M on the first line which is then followed by M strings one per line (formed by letters in range A to Z, upper case only).

For boundaries, we will have 0 < M <= 100 and the length of any string will be less than 100.

## Output

The output consists of one line for each case, indicating the number of pairs of anagrams.

### Example Input

```
2
7
BOBCAT
EXAMPLE
BOCAT
TOBBAC
CABBOT
AMPLEEX
BOBEXAM
10
TEST
TESTS
SETS
TTES
STET
STETS
SSET
SSEET
TESS
TEESS
```

### Example Output

```
4
8
```

### Example Explanation

Note that the first case has the following four anagram pairs:

1. BOBCAT and TOBBAC
2. BOBCAT and CABBOT
3. TOBBAC and CABBOT
4. EXAMPLE and AMPLEEX

And the second case has the following eight anagram pairs:

1. TEST and TTES
2. TEST and STET
3. TTES and STET
4. SETS and SSET
5. SETS and TESS
6. SSET and TESS
7. TESTS and STETS
8. SSEET and TEESS

# Problem 5: Anagrams Triplets (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submission 9 ★
Submitted on 4/20/2021 3:39 PM by Christian Duncan

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **Anagram.java**.

## Description

Two strings are anagrams if you can shuffle the letters in one string and get the other string. For example, DORMITORY and DIRTYROOM are anagrams. Given a list of strings, count how many **triplets** of strings are anagrams of each other.

This problem is identical to Problem 4 except that we are looking at **triplets** instead of pairs, the **range has increased,** and **efficiency** is now important. Your program must complete the test case in **under 60 seconds**.

## Input

The first line contains the number N of cases with 0 < N <= **5**. (Yes, fewer cases.)
This is then followed by N cases. Each case starts with a number M on the first line which is then followed by M strings one per line (formed by letters in range A to Z, upper case only).

For boundaries, we will have 0 < M <= **10000** and the length of any string will be less than **1000**.

## Output

The output consists of one line for each case, indicating the number of pairs of anagrams.

### Example Input

```
2
7
BOBCAT
EXAMPLE
BOCAT
TOBBAC
CABBOT
AMPLEEX
BOBEXAM
11
TEST
TESTS
SETS
TTES
STET
STETS
SSET
SSEET
TESS
TEESS
TTSE
```

### Example Output

```
1
5
```

### Example Explanation

Note that the first case has one one anagram triplet:

1. BOBCAT, TOBBAC, and CABBOT
2. EXAMPLE and AMPLEEX are pairs but there isn't a third element.

And the second case has the following five anagram triplets:

1. TEST, TTES, and STET
2. TEST, TTES, and TTSE
3. TEST, STET, and TTSE
4. TTES, STET, and TTSE
5. SETS, SSET, and TESS
6. TESTS and STETS form a pair but not a triplet
7. SSEET and TEESS form a pair but not a triplet

# Problem 6: The Queen's Jewels (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **Jewels.java**.

## Description

For Her Majesty Queen Elizabeth II's 95th birthday (on April 21), your team has been tasked with helping to design a beautiful crown of jewels for her to display during the various formal ceremonies. The task at hand is to pick out a set of jewels from the royal family's immense private collection. There are a few constraints placed on you:

1. The jewels have values (specified in British pounds, £)
2. The jewels have various weights (measured in carats)
3. The jewels have colors and this set must be only from the British colors - RED, WHITE, and BLUE
4. The crown has a maximum weight capacity (maximum total carats to display)
5. There must be an equal weight of RED, WHITE, and BLUE jewels. We want color balance! If there aren't enough jewels of a specific color then the other colors would have to use this same maximum capacity.
6. Your task it to pick a sample that yields the greatest value while sticking to the constraints - worthy of the royal crown.

In addition, your team has an expert jeweler who can cut a jewel to a proper size if needed to fit within the crown's capacity. The value of a cut jewel is proportional to its original size. For instance, if a jewel were 10 carats and worth 1000£, then cutting it to two carats (20% of original size) would make the value of the cut portion 200£.

Suppose that the collection contains the following jewels:

- Red jewels:
  1. 10£ and 2 carats
  2. 20£ and 3 carats
  3. 18£ and 4 carats
- White jewels:
  1. 8£ and 1.2 carats
  2. 15£ and 3 carats
  3. 6£ and 2 carats
  4. 8£ and 0.4 carats
- Blue jewels:
  1. 4£ and 0.5 carats
  2. 3£ and 0.9 carats
  3. 20£ and 5 carats

Note: This is just a sample. The royal collection is significantly larger and more valuable.

Suppose that the desired capacity for each color is 5 carats. Then the following set yields a maximum value of 84£, rounded to the nearest pound. (4.5 rounds up to 5.)

- Red jewels: Select jewels 1 and 2, for a capacity of 5 and value 30£.
- White jewels: Select jewels 1, 2, and 4, for a capacity of 4.6 and a value of 31£. Then cut (and use) 0.4 carats of jewel 3 yielding a value of 1.2£. This produces a total of 5 carats and a value of 32.2£.
- Blue jewels: Select jewel 1 and 4.5 carats of jewel 3, for a capacity of 5 carats and a value of 4£+18£ = 22£.

Adding up the values of the three jewel types yields 84.2£, which we round to 84£.

Suppose instead that the desired capacity for each color is 7 carats. There are only 6.4 total carats for the blue jewels. Therefore, for balance, the best we can accomplish is 6.4 total carats for each type. The following set yields a maximum value of 100£.

- Red jewels: Select jewels 1 and 2 and 1.4 carats of jewel 3 for 6.4 total carats and a value of 10+20+6.3 = 36.3£.
- White jewels: Select jewels 1, 2, 4, and 1.8 carats of jewel 3 for 6.4 total carats and a value of 8+15+8+5.4 = 36.4£.
- Blue jewels: Naturally, use all of them for a total of 27£.

Adding up the values of the three jewel types yields 99.7£, which we round to 100£.

## Input

The first line contains the number N of cases with 0 < N <= 50.
This is then followed by N cases. Each case starts with a number M on the first line indicating the number of jewels in the collection to choose from. The second line contains the weight capacity (in carats, as an integer value). This is then followed by M lines representing a different jewel. Each such line has three values separated by a single space. The first is the value (in British pounds, as an integer value). The second is the weight of the jewel (in carats, as a value up to two decimal places out - example, 3.12). The third line is the color of the jewel - either RED, WHITE, or BLUE.

For boundaries, the number of jewels will range from 1 to 500 (0 < M <= 500), the value of any individual jewel ranges from 1 to 100000 pounds (in whole numbers), and the weight of any jewel ranges from 0.01 to 100 carats (in increments of 0.01). The weight capacity can range from 1 to 1000000 carats (that is one extremely heavy crown).

## Output

The output consists of one line for each case, indicating the maximum value possible reported in British pounds and rounded to the nearest whole pound.

### Example Input

```
2
10
5
10 2 RED
8 1.2 WHITE
4 0.5 BLUE
```

of jewels from the royal family's immense private collection. There are a few constraints placed on you:

1. The jewels have values (specified in British pounds, £)
2. The jewels have various weights (measured in carats)
3. The jewels have colors and this set must be only from the British colors - RED, WHITE, and BLUE
4. The crown has a maximum weight capacity (maximum total carats to display)
5. There must be an equal weight of RED, WHITE, and BLUE jewels. We want color balance! If there aren't enough jewels of a specific color then the other colors would have to use this same maximum capacity.
6. Your task it to pick a sample that yields the greatest value while sticking to the constraints - worthy of the royal crown.

In addition, your team has an expert jeweler who can cut a jewel to a proper size if needed to fit within the crown's capacity. The value of a cut jewel is proportional to its original size. For instance, if a jewel were 10 carats and worth 1000£, then cutting it to two carats (20% of original size) would make the value of the cut portion 200£.

Suppose that the collection contains the following jewels:

- Red jewels:
    1. 10£ and 2 carats
    2. 20£ and 3 carats
    3. 18£ and 4 carats
- White jewels:
    1. 8£ and 1.2 carats
    2. 15£ and 3 carats
    3. 6£ and 2 carats
    4. 8£ and 0.4 carats
- Blue jewels:
    1. 4£ and 0.5 carats
    2. 3£ and 0.9 carats
    3. 20£ and 5 carats

Note: This is just a sample. The royal collection is significantly larger and more valuable.

Suppose that the desired capacity for each color is 5 carats. Then the following set yields a maximum value of 84£, rounded to the nearest pound. (4.5 rounds up to 5.)

- Red jewels: Select jewels 1 and 2, for a capacity of 5 and value 30£.
- White jewels: Select jewels 1, 2, and 4, for a capacity of 4.6 and a value of 31£. Then cut (and use) 0.4 carats of jewel 3 yielding a value of 1.2£. This produces a total of 5 carats and a value of 32.2£.
- Blue jewels: Select jewel 1 and 4.5 carats of jewel 3, for a capacity of 5 carats and a value of 4£+18£ = 22£.

Adding up the values of the three jewel types yields 84.2£, which we round to 84£.

Suppose instead that the desired capacity for each color is 7 carats. There are only 6.4 total carats for the blue jewels. Therefore, for balance, the best we can accomplish is 6.4 total carats for each type. The following set yields a maximum value of 100£.

- Red jewels: Select jewels 1 and 2 and 1.4 carats of jewel 3 for 6.4 total carats and a value of 10+20+6.3 = 36.3£.
- White jewels: Select jewels 1, 2, 4, and 1.8 carats of jewel 3 for 6.4 total carats and a value of 8+15+8+5.4 = 36.4£.
- Blue jewels: Naturally, use all of them for a total of 27£.

Adding up the values of the three jewel types yields 99.7£, which we round to 100£.

## Input

The first line contains the number N of cases with 0 < N <= 50.
This is then followed by N cases. Each case starts with a number M on the first line indicating the number of jewels in the collection to choose from. The second line contains the weight capacity (in carats, as an integer value). This is then followed by M lines representing a different jewel. Each such line has three values separated by a single space. The first is the value (in British pounds, as an integer value). The second is the weight of the jewel (in carats, as a value up to two decimal places out - example, 3.12). The third line is the color of the jewel - either RED, WHITE, or BLUE.

For boundaries, the number of jewels will range from 1 to 500 (0 < M <= 500), the value of any individual jewel ranges from 1 to 100000 pounds (in whole numbers), and the weight of any jewel ranges from 0.01 to 100 carats (in increments of 0.01). The weight capacity can range from 1 to 1000000 carats (that is one extremely heavy crown).

## Output

The output consists of one line for each case, indicating the maximum value possible reported in British pounds and rounded to the nearest whole pound.

### Example Input

```
2
10
5
10 2 RED
8 1.2 WHITE
4 0.5 BLUE
15 3 WHITE
3 0.9 BLUE
20 5 BLUE
6 2 WHITE
20 3 RED
18 4 RED
8 0.4 WHITE
10
7
10 2 RED
8 1.2 WHITE
4 0.5 BLUE
15 3 WHITE
3 0.9 BLUE
20 5 BLUE
6 2 WHITE
20 3 RED
18 4 RED
8 0.4 WHITE
```

### Example Output

```
84
100
```

# Problem 7: The Queen's Jewels Revised (Java) OPEN

Edit

| OPEN DATE | DUE DATE | SUBMISSIONS | GRADING STRATEGY |
|---|---|---|---|
| 4/21/2021 9:00 AM | 4/21/2021 11:00 AM | Unlimited | Best Submission |



HIGHEST SUBMISSION

View All Instructor/TA Submissions

Submit Project

ACTIONS

Submit

Open in Mimir IDE

DESCRIPTION

## File

For this problem, be sure your main file is called **Jewels.java**.

## Description

For Her Majesty Queen Elizabeth II's 95th birthday (on April 21), your team has been tasked with helping to design a beautiful crown of jewels for her to display during the various formal ceremonies. The task at hand is to pick out a set of jewels from the royal family's immense private collection. There are a few constraints placed on you:

1. The jewels have values (specified in British pounds, £)
2. The jewels have various weights (measured in carats)
3. The jewels have colors and this set must be only from the British colors - RED, WHITE, and BLUE
4. The crown has a maximum weight capacity (maximum total carats to display)
5. There must be an equal weight of RED, WHITE, and BLUE jewels. We want color balance! If there aren't enough jewels of a specific color then the other colors would have to use this same maximum capacity.
6. Your task it to pick a sample that yields the greatest value while sticking to the constraints - worthy of the royal crown.

**You have been told in no uncertain terms that the jewels are not allowed to be cut! You must either use the jewel or not. This restriction means sometimes it will not even be possible to achieve a perfect color balance (other than not using any jewels at all). But you decide to do your best to see if the order can be satisfied.**

After some frustrating exchanges, you have agreed to only choose among a select set of jewels and measure the jewels capacity in terms of 10 carat increments (10, 20, 30, ...). In addition, you will only design the main focal point of the crown and produce a much more pleasing but modest capacity of at most 200 total carats per color type, again measured in multiples of 10. The final product can throw in lesser valued jewels if so desired.

Suppose that the collection contains the following jewels:

- Red jewels:
  1. 5110£ and 20 carats
  2. 6055£ and 30 carats
  3. 7200£ and 60 carats
- White jewels:
  1. 24200£ and 50 carats
  2. 8015£ and 20 carats
  3. 4204£ and 10 carats
  4. 8211£ and 20 carats
- Blue jewels:
  1. 15000£ and 10 carats
  2. 90000£ and 70 carats
  3. 2500£ and 20 carats

Note: This is just a sample. The royal collection is significantly larger and more valuable.

Suppose that the desired capacity for each color is 100 carats. This is not completely achievable but there is a solution with 80 carats per color with total value 153925£. There is a solution with 90 carats for each color but that has less value (146181£).

- Red jewels: Select jewels 1 and 3, for a capacity of 80 and value 12310£.
- White jewels: Select jewels 1, 3, and 4, for a capacity of 80 and a value of 36615£.
- Blue jewels: Select jewel 1 and 2, for a capacity of 80 and a value of 105000£.

Adding up the values of the three jewel types yields 153925£.

Sometimes, no selection will work. For instance, if the desired capacity was 10 carats, there is no selection of red jewels that can achieve this - since they are all larger than 10 carats. (There can be other reasons as well!)

## Input

The first line contains the number N of cases with 0 < N <= 50.
This is then followed by N cases. Each case starts with a number M on the first line indicating the number of jewels in the collection to choose from. The second line contains the weight capacity (in carats, as an integer value and a multiple of ten). This is then followed by M lines representing a different jewel. Each such line has three values separated by a single space. The first is the value (in British pounds, as an integer value). The second is the weight of the jewel (in carats, as a multiple of ten). The third line is the color of the jewel - either RED, WHITE, or BLUE.

For boundaries, the number of jewels will range from 1 to 30 (0 < M <= 30), the value of any individual jewel ranges from 1 to 100000 pounds (in whole numbers), and the weight of any jewel ranges from 10 to 200 carats (in increments of 10). The weight capacity can range from 10 to 200 carats in increments of 10.

## Output

The output consists of one line for each case, indicating the maximum value possible reported in British pounds and rounded to the nearest whole pound. If it isn't possible to select any jewels, NOT POSSIBLE should be output.

### Example Input

```
3
10
90
5110 20 RED
24200 50 WHITE
15000 10 BLUE
6055 30 RED
```

jewels capacity in terms of 10 carat increments (10, 20, 30, ...). In addition, you will only design the main focal point of the crown and produce a much more pleasing but modest capacity of at most 200 total carats per color type, again measured in multiples of 10. The final product can throw in lesser valued jewels if so desired.

Suppose that the collection contains the following jewels:

- Red jewels:
    1. 5110£ and 20 carats
    2. 6055£ and 30 carats
    3. 7200£ and 60 carats
- White jewels:
    1. 24200£ and 50 carats
    2. 8015£ and 20 carats
    3. 4204£ and 10 carats
    4. 8211£ and 20 carats
- Blue jewels:
    1. 15000£ and 10 carats
    2. 90000£ and 70 carats
    3. 2500£ and 20 carats

Note: This is just a sample. The royal collection is significantly larger and more valuable.

Suppose that the desired capacity for each color is 100 carats. This is not completely achievable but there is a solution with 80 carats per color with total value 153925£. There is a solution with 90 carats for each color but that has less value (146181£).

- Red jewels: Select jewels 1 and 3, for a capacity of 80 and value 12310£.
- White jewels: Select jewels 1, 3, and 4, for a capacity of 80 and a value of 36615£.
- Blue jewels: Select jewel 1 and 2, for a capacity of 80 and a value of 105000£.

Adding up the values of the three jewel types yields 153925£.

Sometimes, no selection will work. For instance, if the desired capacity was 10 carats, there is no selection of red jewels that can achieve this - since they are all larger than 10 carats. (There can be other reasons as well!)

## Input

The first line contains the number N of cases with 0 < N <= 50.
This is then followed by N cases. Each case starts with a number M on the first line indicating the number of jewels in the collection to choose from. The second line contains the weight capacity (in carats, as an integer value and a multiple of ten). This is then followed by M lines representing a different jewel. Each such line has three values separated by a single space. The first is the value (in British pounds, as an integer value). The second is the weight of the jewel (in carats, as a multiple of ten). The third line is the color of the jewel - either RED, WHITE, or BLUE.

For boundaries, the number of jewels will range from 1 to 30 (0 < M <= 30), the value of any individual jewel ranges from 1 to 100000 pounds (in whole numbers), and the weight of any jewel ranges from 10 to 200 carats (in increments of 10). The weight capacity can range from 10 to 200 carats in increments of 10.

## Output

The output consists of one line for each case, indicating the maximum value possible reported in British pounds and rounded to the nearest whole pound. If it isn't possible to select any jewels, NOT POSSIBLE should be output.

### Example Input

```
3
10
90
5110 20 RED
24200 50 WHITE
15000 10 BLUE
6055 30 RED

8015 20 WHITE
90000 70 BLUE
7200 60 RED
4204 10 WHITE
8211 20 WHITE
2500 20 BLUE
10
10
5110 20 RED
24200 50 WHITE
15000 10 BLUE
6055 30 RED
8015 20 WHITE
90000 70 BLUE
7200 60 RED
4204 10 WHITE
8211 20 WHITE
2500 20 BLUE
15
200
30000 10 RED
1500 90 RED
70000 40 RED
85000 80 RED
12000 160 RED
40000 10 WHITE
1000 90 WHITE
20000 40 WHITE
42000 80 WHITE
90000 160 WHITE
10000 10 BLUE
2000 90 BLUE
15000 40 BLUE
62000 80 BLUE
22000 160 BLUE
```

### Example Output

```
153925
NOT POSSIBLE
374000
```

# Problem 8: Contact Tracing (Java) OPEN



DESCRIPTION

## File

For this problem, be sure your main file is called **ContactTracer.java**.

## Description

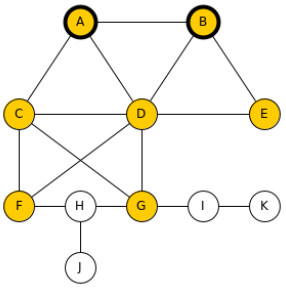
Contact tracing is a critical step in preventing the spread of any illness, especially during a global pandemic. But how exactly does one perform contact tracing? There are many steps involved. But here we will focus on one key step.

Suppose you are given a list of contacts between individuals. (This data could be collected for example by proximity sensors on your phone.) And then suppose you are given a list of individuals that have been found to be infected. You would then look at the individuals that have come into contact with those individuals,
and perhaps go another level down to those individuals, and so on, for a given distance D interactions away.
That list would form the list of individuals that were likely exposed.

We are of course oversimplifying the task. There are other factors such as time spent in contact, proximity to the individuals, and time since the last contact.

Your task is to design, develop, and implement an efficient program that can determine how many individuals have been exposed (including the infected individuals as well). Ideally, we would have you report the list of individuals but for simplicity in scoring, it is easier to just state the total number.

Here is an example graph of various known interactions.
The two heavily highlighted nodes are the two known infections and the other highlighted nodes are the individuals that are at most 2 contacts away.




## Input

The first line contains the number N of cases with 0 < N <= 100. This is then followed by N cases. Each case consists of an undirected graph G followed by the list of infected individuals and a distance value D.

More specifically, for each case,

1. The first line of the input consists of an integer N representing the number of individuals.
2. This is then followed by N lines representing the unique identifier of each individual. Each identifier consists of letters A-Z or a-z or digits 0-9. (In our example, we will just use "names.") But, no two individuals have the same id.
3. The next line of the input consists of an integer M represening the number of known interactions.
4. This is then followed by M pairs of identifiers A and B, meaning A and B had contact with each other.
5. The next line of the input consists of two numbers K, the number of infected individuals, and D, the distance to calculate away from any infected individual.
6. This is then followed by K identifiers representing each of the known infected individuals.

Here are some assumptions that you can make:

1. You can assume that the number of vertices is no more than 1000.
2. You can assume that the contact graph provided is valid and in the valid format (no incorrect input format to check).
3. You can assume that there are no duplicate interactions or simple loops (A interacting with A).

## Output

The output consists of one line for each case, that is the number individuals (include the original infected persons) that are distance at most D away from the known infected individuals.

### Example Input

```
2
11
Alice
Bob
Clara
David
Eve
Fred
Georgia
```

1. You can assume that the number of vertices is no more than 1000.
2. You can assume that the contact graph provided is valid and in the valid format (no incorrect input format to check).
3. You can assume that there are no duplicate interactions or simple loops (A interacting with A).

## Output

The output consists of one line for each case, that is the number individuals (include the original infected persons) that are distance at most D away from the known infected individuals.

### Example Input

```
2
11
Alice
Bob
Clara
David
Eve
Fred
Georgia
Heath
Ingrid
Jason
Kathy
16
Bob Alice
Alice Clara
Alice David
Bob David
Eve Bob
Clara David
Georgia Clara
Fred Clara
David Fred
Eve David
David Georgia
Fred Heath
Heath Georgia
Ingrid Georgia
Ingrid Kathy
Jason Heath
2 2
Bob
Alice
20
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
C1
C2
C3
C4
C5
D1
D2
D3
D4
D5
25
A1 A2
A2 A3
A3 A4
A4 A5
A5 B1
B1 B2
B2 B3
B3 B4
B4 B5
B5 C1
C1 C2
C2 C3
C3 C4
C4 C5
C5 A1
A3 D1
A4 D1
B3 D1
B4 D1
C3 D1
C4 D1
D1 D2
D1 D3
D1 D4
D1 D5
1 2
D1
```

### Example Output

```
7
17
```