Lab 3A

Problem 1: Bob and Grades

Bob is a very studious child but messed up his latest quiz due to an unprecedented event. To avoid losing his grades at any cost, he has found an interesting way to maximize his scores: by stealing marks from others. He has obtained two lists, containing, all the marks and now wants to steal as much as he can. However, he can only make one modification to his score before it becomes too suspicious for the professor to detect.

Given Bob's initial score and two arrays A and B of size 20 with all the marks of the students, Bob can perform upto any one of the 2 types of queries ONCE

- Q1 : Take an element from array A and add to the grade
- Q2 : Take an element from array B and add half of it to his grade

Return the maximum score Bob can achieve, in Binary Notation

Input:

- First line containing one integer, Bob's Initial Score
- Second and Third line containing 20 space seperated values, The score arrays

Output:

- A single line containing value of the maximum possible grade, in Binary.

Constraints:

- N = 20
- $0 \leq$ `A[i],B[i]` ≤ 4000

Good luck!

Example:

Input:

```
145
791 19 406 505 896 79 670 398 198 880 581 639 364 131 519 503 245 924 640 438
255 966 333 783 335 954 189 541 719 863 219 778 360 168 479 562 345 338 884 209
```

Copy

Output:

```
10000101101
```

Explanation:

Here, we choose to add 924 from the first array, making the final score as 1069 (10000101101 in Binary)

Input:

```
319
237 733 244 110 448 678 964 717 98 769 144 645 817 179 163 932 320 983 858 93
458 768 599 830 879 461 555 827 427 476 935 2 720 10 849 418 641 435 918 536
```

Copy

Output:

```
10100010110
```

Explanation:

Here, we chose to add 983 from the first array, making the final score as 1302 (10100010110 in Binary)

Problem 2: Terrific Escape

While escaping the Prof's Lab, Bob stumbled upon the security system, which he now needs to bypass in order to get away, To get through this security system, you have to decode a password from the Security Matrix (2D Array) of size NxN.

The sum of each row in this matrix represents a Codeword, The password will be the amount of SIMILAR pair of codewords there are.

Two codewords are said to be SIMILAR, if they are

- Equal in value
- Coprime

Note: Coprime numbers are those numbers that have only one common factor, namely 1.

Input:

- First line containing one integer, Size of the Security Matrix
- Followed by N lines containing N space-seperated values

Output:

- A single line containing the password

Constraints:

- N ≤ 100
- -1000 ≤ `A[i][j]` ≤ 1000

Input:

```
2
1 2
3 4
```

Copy

Output:

```
1
```

Explanation:

First and second codewords would be 3 and 7, which are co-prime, so total SIMILAR pairs (Row 1 and 2), is 1

Input:

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

Copy

Output:

```
3
```

Explanation:

Here

- Row 1 and 2, They have an equal sum (5)
- Row 1 and 3, They are co-prime (5 and 12 are co-prime)
- Row 2 and 3, They are co-prime (5 and 12 are co-prime)

Problem 3: Bob's Path Back Home

Bob has successfully made his way out of the lab but now needs to get home, for this he needs to cross a grassy field represented by a matrix of size NxM

Each cell in the matrix contains a number that represents the energy Bob will spend if he steps on that cell. Bob starts at the top-left corner of the matrix and needs to reach the bottom-right corner.

Bob can only move right or down at each step. The goal is to find the total energy Bob will spend if he takes the path where he always chooses the cell with the least energy. (If both have the same energy, he will choose to move rightwards)

Note: His initial energy is the cell he starts on

Input Format:

First Line containing 2 space-seperated values `N,M`.

Followed by, N lines, each containing M space-seperated values

Due to possible difficulty faced in input of matrices, we've added it in the handout for this lab, but you are expected to do it yourself from next lab onwards.

Output Format:

- Single line containing the total energy spent by Bob

Constraints:

- N,M ≤ 1000
- -10000 ≤ `A[i][j]` ≤ 10000

Examples:

Input:

```
3 3
1 2 3
4 5 6
7 8 9
```

Copy

Output:

```
21
```

Explanation:

You start with intial energy as 1, now we choose 2 (out of 2,4), now choose 3 (out of 3,5), then 6, then 9 (there's no right option after reaching 3)

1 + 2 + 3 + 6 + 9 = 21

Submission Guidelines

- Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.