

Shivam , Shantam and their Absolute Difference - Hackerearth

Tuesday, May 19, 2020 7:53 PM

Shivam and Shantam are really good friends. What mostly brings them together is the hours of group study they have done in the two years of their engineering, and there is only more to add to this. They both start studying together for exams , and keep explaining topics and clearing each other's slightest of doubts.

But , unfortunately , marks and money can always make or break relations. Both of them have to answer N questions in tomorrow's paper. We also have two arrays , A and B . A_i denotes the marks Shivam will get if he answers the i^{th} question. B_i denotes the marks Shantam will get if he answers the i^{th} question. The person not choosing to answer the question would get no marks.

Obviously , there are many ways in which they can give their papers. Let the final score of Shivam be S and that of Shantam be K . We are interested in knowing the number of ways such that the absolute difference in S and K is less than or equal to Q .

Input Format :

The first line contains N , the number of questions. The next line contains N space separated integers denoting A_i , the marks that Shivam can score if he attempts the i^{th} question. The next line contains N space separated integers denoting B_i , the marks that Shantam can score if he attempts the i^{th} question. The next line contains Q , the maximum allowed absolute difference.

Output Format :

Since the number of ways can be huge , output the answer modulo $10^9 + 7$ in a separate line.

Constraints :

- $1 \leq N \leq 200$
- $0 \leq A_i, B_i \leq 2 * 10^3$
- $0 \leq Q \leq 10^6$

SAMPLE INPUT

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

SAMPLE OUTPUT

```
6
```

Explanation

There are 6 ways to get an absolute difference less than equal to 2 :

1. Shivam : {1 , 0} Shantam: {0 , 0} i. e. Shivam answers only the first question, but Shantam answers none.

2. Shivam : {0 , 2} Shantam: {0 , 0} i. e. Shivam answers only the second question, but Shantam answers none.
3. Shivam : {0 , 2} Shantam: {4 , 0}
4. Shivam : {1 , 2} Shantam: {4 , 0}
5. Shivam : {1, 2} Shantam: {0 , 5}
6. Shivam : {0 , 0} Shantam: {0 , 0}

The Code:

```
#include <stdio.h>
#include <stdlib.h>
#define MOD 1000000007
int main(){
    int N, Q, prev, curr;
    long long int ASum, BSum, maxSum, result, diff, other_marks, prod, i, j;
    int A[200];
    int B[200];
    long long int* A_marks[2];
    A_marks[0] = (long long int*)calloc(1000000, sizeof(long long int));
    A_marks[1] = (long long int*)calloc(1000000, sizeof(long long int));
    long long int* B_marks[2];
    B_marks[0] = (long long int*)calloc(1000000, sizeof(long long int));
    B_marks[1] = (long long int*)calloc(1000000, sizeof(long long int));
    scanf("%d", &N);
    ASum = 0;
    for(i=0; i<N; i++) {
        scanf("%d", &A[i]);
        ASum += A[i];
    }
    BSum = 0;
    for(i=0; i<N; i++) {
        scanf("%d", &B[i]);
        BSum += B[i];
    }
    maxSum = ASum > BSum ? ASum : BSum;
    A_marks[0][0] = 1; A_marks[1][0] = 1;
    B_marks[0][0] = 1; B_marks[1][0] = 1;
    prev = 0; curr = 0;
    for(i=0; i<N; i++) {
        prev = (i % 2);
        curr = 1-prev;
        for(j=A[i]; j<=ASum; j++) {
            A_marks[curr][j] = (A_marks[prev][j] + A_marks[prev][j-A[i]]) % MOD;
        }
        for(j=0; j<A[i]; j++) {
            A_marks[curr][j] = A_marks[prev][j];
        }

        for(j=B[i]; j<=BSum; j++) {
            B_marks[curr][j] = (B_marks[prev][j] + B_marks[prev][j-B[i]]) % MOD;
        }
        for(j=0; j<B[i]; j++) {
            B_marks[curr][j] = B_marks[prev][j];
        }
    }
}
```

```

    }

    for(i=1; i<=maxSum; i++) {
        A_marks[curr][i] = (A_marks[curr][i] + A_marks[curr][i-1]) % MOD;
        B_marks[curr][i] = (B_marks[curr][i] + B_marks[curr][i-1]) % MOD;
    }
    scanf("%d", &Q);
    result = 1;
    for(i=1; i<=maxSum; i++) {
        if(A_marks[curr][i] != A_marks[curr][i-1]) {
            diff = (i-Q-1) < 0 ? 0 : B_marks[curr][i-Q-1];
            other_marks = (B_marks[curr][i] - diff + MOD) % MOD;
            diff = (A_marks[curr][i] - A_marks[curr][i-1] + MOD) % MOD;
            result += (long long int)diff * other_marks % MOD;
            result %= MOD;
        }
        if(B_marks[curr][i] != B_marks[curr][i-1]) {
            diff = (i-Q-1) < 0 ? 0 : A_marks[curr][i-Q-1];
            other_marks = (A_marks[curr][i-1] - diff + MOD) % MOD;
            diff = (B_marks[curr][i] - B_marks[curr][i-1] + MOD) % MOD;
            result += (long long int)diff * other_marks % MOD;
            result %= MOD;
        }
    }
    printf("%d\n", result);
}

```

The Stats:

Score

30.0

Time (sec)

1.82325

Memory (KiB)

31580

Language

C