1843 - I2P (I) 2019_Yang_CS_practice_M2 (/contest/edit/1843/)

Scoreboard (/contest/scoreboard/1843/)

Time

2019/11/19 21:00:00 | 20days, 13:28:08 | 2019/12/10 12:00:00

Clarification

#	Problem	Asker	Description	Reply	Replier	Reply Time	For all team

Clarify

Overview | Problem

11241 - Simple Addition

(/problem/11241/edit/) Status (/status/?pid=11241) | Limits Submit (/users/submit/11241)

Description

Four arrays A, B, C and D with size MxN. B, C, D are stored in an array of pointers, and A is stored in a separated array. The task is to add the designated elements of A and the corresponding element from a chosen array of B, C, D, and output the result.

Take sample input as an example. Size of the arrays are 3 rows by 2 columns, initial values are as the form below. Input number “2” in the sixth line indicates D is chosen array and we want the sum of elements with two indices (1,0), (2,0)

A

0	1
2	3
4	5

B

0	1
4	9
16	25

C

0	1
2	0
1	2

D

0	2
8	7
10	6

Therefore, the answer is A(1,0)+D(1,0)+A(2,0)+D(2,0)=2+8+4+10=24

You will be provided with the following sample code, and asked to implement function "addition".

```
#include <stdio.h>

int addition(int*, int, int*[], int*, int);

int main(void) {

    int a[50][50], b[50][50], c[50][50], d[50][50];
    int index_to_add[20];
    int *entry[3];
    int i, j, m, n, array_num, num_ind;

    scanf("%d %d", &m, &n);

    for(i=0; i<m; i++)
```

```
        for(j=0; j<n; j++)
            scanf("%d", &a[i][j]);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            scanf("%d", &b[i][j]);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            scanf("%d", &c[i][j]);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            scanf("%d", &d[i][j]);

    scanf("%d", &array_num);
    scanf("%d", &num_ind);
    for(i=0; i<num_ind*2; i=i+2)
        scanf("%d %d", &index_to_add[i], &index_to_add[i+1]);

    entry[0] = &b[0][0];
    entry[1] = &c[0][0];
    entry[2] = &d[0][0];

    printf("%d\n", addition(&a[0][0], array_num, entry, index_to_add, num_ind));

    return 0;
}

int addition(int* ptr_a, int array_num, int* entry[], int* index_to_add, int num_ind){
    /*your code*/
}
```

Input

The first line is the size of arrays, **M** rows by **N** columns. The following four lines are initial values arranged by row major. The sixth line tells B, C or D is chose. The next line is the number of elements **k** to be summed. And the last **k** lines are the row and column indices of elements to be add. 0 < M, N < 50. 0 < k < 10. Index starts from 0.

Output

An integer, sum of desired elements.

Sample Input

Download (data:text/plain;charset=utf-8,3%202%0A0%201%202%203%204%205%0A0%201%204%209%2016%2025%0A0%201%202%200%201%202%0A0%202%208%207%2010%206%

```
3 2
0 1 2 3 4 5
0 1 4 9 16 25
0 1 2 0 1 2
0 2 8 7 10 6
2
2
1 0
2 0
```

Sample Output

Download (data:text/plain;charset=utf-8,24%0A)

```
24
```

Discuss