#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\*\* split\_string(char\*);

// Complete the countApplesAndOranges function below.

void countApplesAndOranges(int s, int t, int a, int b, int apples\_count, int\* apples, int oranges\_count, int\* oranges) {

int appcount,orangecount;

appcount =0;

orangecount =0;

int temp;

int i;

for(i=0;i<apples\_count;i++)

{

if(apples[i] >0)

{

temp = apples[i] + a;

if(temp>=s && temp<=t )

{

appcount++;

}

}

}

for(i=0;i<oranges\_count;i++)

{

if(oranges[i]<0)

{

temp = oranges[i] + b;

if(temp>=s && temp<=t)

{

orangecount++;

}

}

}

printf("%d",appcount);

printf("\n%d",orangecount);

}

int main()

{

char\*\* st = split\_string(readline());

char\* s\_endptr;

char\* s\_str = st[0];

int s = strtol(s\_str, &s\_endptr, 10);

if (s\_endptr == s\_str || \*s\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\* t\_endptr;

char\* t\_str = st[1];

int t = strtol(t\_str, &t\_endptr, 10);

if (t\_endptr == t\_str || \*t\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\*\* ab = split\_string(readline());

char\* a\_endptr;

char\* a\_str = ab[0];

int a = strtol(a\_str, &a\_endptr, 10);

if (a\_endptr == a\_str || \*a\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\* b\_endptr;

char\* b\_str = ab[1];

int b = strtol(b\_str, &b\_endptr, 10);

if (b\_endptr == b\_str || \*b\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\*\* mn = split\_string(readline());

char\* m\_endptr;

char\* m\_str = mn[0];

int m = strtol(m\_str, &m\_endptr, 10);

if (m\_endptr == m\_str || \*m\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\* n\_endptr;

char\* n\_str = mn[1];

int n = strtol(n\_str, &n\_endptr, 10);

if (n\_endptr == n\_str || \*n\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\*\* apples\_temp = split\_string(readline());

int\* apples = malloc(m \* sizeof(int));

for (int i = 0; i < m; i++) {

char\* apples\_item\_endptr;

char\* apples\_item\_str = \*(apples\_temp + i);

int apples\_item = strtol(apples\_item\_str, &apples\_item\_endptr, 10);

if (apples\_item\_endptr == apples\_item\_str || \*apples\_item\_endptr != '\0') { exit(EXIT\_FAILURE); }

\*(apples + i) = apples\_item;

}

int apples\_count = m;

char\*\* oranges\_temp = split\_string(readline());

int\* oranges = malloc(n \* sizeof(int));

for (int i = 0; i < n; i++) {

char\* oranges\_item\_endptr;

char\* oranges\_item\_str = \*(oranges\_temp + i);

int oranges\_item = strtol(oranges\_item\_str, &oranges\_item\_endptr, 10);

if (oranges\_item\_endptr == oranges\_item\_str || \*oranges\_item\_endptr != '\0') { exit(EXIT\_FAILURE); }

\*(oranges + i) = oranges\_item;

}

int oranges\_count = n;

countApplesAndOranges(s, t, a, b, apples\_count, apples, oranges\_count, oranges);

return 0;

}

char\* readline() {

size\_t alloc\_length = 1024;

size\_t data\_length = 0;

char\* data = malloc(alloc\_length);

while (true) {

char\* cursor = data + data\_length;

char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

if (!line) { break; }

data\_length += strlen(cursor);

if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') { break; }

size\_t new\_length = alloc\_length << 1;

data = realloc(data, new\_length);

if (!data) { break; }

alloc\_length = new\_length;

}

if (data[data\_length - 1] == '\n') {

data[data\_length - 1] = '\0';

}

data = realloc(data, data\_length);

return data;

}

char\*\* split\_string(char\* str) {

char\*\* splits = NULL;

char\* token = strtok(str, " ");

int spaces = 0;

while (token) {

splits = realloc(splits, sizeof(char\*) \* ++spaces);

if (!splits) {

return splits;

}

splits[spaces - 1] = token;

token = strtok(NULL, " ");

}

return splits;

}