#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 squares function below.

int squares(int a, int b) {

double result1;

double result2;

int a1,a2;

long double left1,left2;

result1 = sqrt((double)a);

result2 = sqrt((double)b);

a1 = (int) result1;

a2= (int) result2;

left1 = (long double)result1 - (long double) a1;

left2 = (long double)result2 - (long double) a2;

if(left1 > 0 && a1>0)

{

a1 = a1+1;

}

if(a2-a1 ==0)

{

return 0;

}

return a2-a1+1;

}

int main()

{

FILE\* fptr = fopen(getenv("OUTPUT\_PATH"), "w");

char\* q\_endptr;

char\* q\_str = readline();

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

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

for (int q\_itr = 0; q\_itr < q; q\_itr++) {

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); }

int result = squares(a, b);

fprintf(fptr, "%d\n", result);

}

fclose(fptr);

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;

}