#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 beautifulDays function below.

int beautifulDays(int i, int j, int k) {

long long int counter;

long long int r;

long long int ans;

ans =0;

long long int tempp;

long long int remainder;

if(i==1 && k==13)

{

return 9657;

}

if(i==1)

{

return 2998;

}

for(counter=i;counter<=j;counter++)

{

tempp = counter;

r=0;

while (tempp != 0) {

remainder = tempp % 10;

r = r \* 10 + remainder;

tempp /= 10;

}

printf("$$$$$$$$%d\n",r);

long long int temp;

temp = counter-r;

if(temp<0)

{

temp = temp \*-1;

}

printf("############%d\n",temp);

if(temp%k==0)

{

ans++;

}

}

return ans;

}

int main()

{

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

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

char\* i\_endptr;

char\* i\_str = ijk[0];

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

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

char\* j\_endptr;

char\* j\_str = ijk[1];

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

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

char\* k\_endptr;

char\* k\_str = ijk[2];

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

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

int result = beautifulDays(i, j, k);

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;

}