#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

/\*

\* Complete the pageCount function below.

\*/

int pageCount(int n, int p) {

if(p==1)

{

return 0;

}

else if(n%2==0 && p==n)

{

return 0;

}

else if(n%2!=0 && (p==n || p==n-1))

{

return 0;

}

else{

long long int front,back;

float fltfront,fltback;

float temp;

float calc;

// calculationg pages from starting

temp = p-1;

fltfront = temp/2;

front = (int) fltfront;

calc = fltfront - front;

if(calc!=0)

{

front = front + 1;

}

//calculating pages from end

if(n%2==0)

{

temp = n - p;

fltback = temp /2;

back = (int) fltback;

calc = fltback - back;

if(calc!=0)

{

back = back + 1;

}

}

if(n%2!=0)

{

temp = n - p;

fltback = temp /2;

back = (int) fltback;

}

if(front>back)

{

return back;

}

else {

return front;

}

}

}

int main()

{

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

char\* n\_endptr;

char\* n\_str = readline();

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

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

char\* p\_endptr;

char\* p\_str = readline();

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

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

int result = pageCount(n, p);

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;

}