const readline = require("readline");

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout,

});

rl.question("Enter N (number of rows): ", function (num) {

const numOfRows = parseInt(num);

rl.question("Enter T (pattern type, 'a' or '1'): ", function (patternType) {

const answer = [];

const firstChar = patternType;

const convertValue = (value) => {

return String.fromCharCode(value);

};

if (isNaN(parseInt(firstChar))) {

const charCode = firstChar.charCodeAt();

let left = 1,

right = charCode + numOfRows - 2;

let top = 0,

bottom = charCode + numOfRows - 2;

for (let i = 0; i < numOfRows; i++) {

const tempArr = [];

for (let j = 0; j < numOfRows; j++) {

if (i === 0) {

tempArr.push(convertValue(charCode + top++));

} else if (j === 0) {

tempArr.push(convertValue(charCode + left++));

} else if (i === numOfRows - 1) {

tempArr.push(convertValue(bottom--));

} else if (j === numOfRows - 1) {

tempArr.push(convertValue(right--));

} else {

tempArr.push(" ");

}

}

answer.push(tempArr);

}

} else {

let l = 2,

r = numOfRows - 1,

t = 1,

b = numOfRows - 1;

for (let i = 0; i < numOfRows; i++) {

const tempArr = [];

for (let j = 0; j < numOfRows; j++) {

if (i === 0) {

tempArr.push(t++);

} else if (j === 0) {

tempArr.push(l++);

} else if (i === numOfRows - 1) {

tempArr.push(b--);

} else if (j === numOfRows - 1) {

tempArr.push(r--);

} else {

tempArr.push(" ");

}

}

answer.push(tempArr);

}

}

// Print the pattern with proper spacing

answer.forEach((row) => {

console.log(

row.map((item) => (item === " " ? " " : `${item} `)).join("")

);

});

rl.close();

});

});



