/\*\* Logic \*/

let baseNumber = 5;

/\*\*

\*

Take Grid and searchPattern from test cases then convert seacrhedpattern to one integer hash value. after converting it, iterate through

Grid matrix and convert also grid element to hash one by one and compare with pattern hashed value within seacrhed pattern range.

\*/

function searchPatternInGrid(gridPatternRows,gridPatternColumns,grid,testCasesRows,testCasesColumns,pattern){

let gridPattern = [];

let testCasePattern = [];

grid.split("\n").forEach((value,index)=>{

gridPattern.push(value.split(""));

})

pattern.split("\n").forEach((value,index)=>{

testCasePattern.push(value.split(""))

})

let matched = 'NO';

let patternHashed = getHashedIntegerfromArray(testCasePattern,0,testCasesRows,0,testCasesColumns);

let maxRowsInGridForMatch = gridPatternRows - testCasesRows;

let maxColumnInGridForMatch = gridPatternColumns - testCasesColumns;

for(k=0;k<=maxRowsInGridForMatch;k++){

for(l=0;l<=maxColumnInGridForMatch;l++){

let getHashedGridSubarray = getHashedIntegerfromArray(gridPattern,k,k+testCasesRows,l,l+testCasesColumns);

if(getHashedGridSubarray == patternHashed){

matched = 'YES';

break;

}

}

if(matched == 'YES'){

break;

}

}

return matched;

}

function getHashedIntegerfromArray(inputArray,fromRows,tillRows,fromColums,tillColumns){

let hash = 0;

for(let i = fromRows; i< tillRows; i++){

for(let j= fromColums; j< tillColumns; j++){

hash = baseNumber \* hash + parseInt(inputArray[i][j]);

}

}

return hash;

}

/\*\* Test Cases \*/

/\* Test-Case 1 - Grid Information like rows, columns. \*\*/

let gridPatternRows1 = 10;

let gridPatternColumns1 = 10;

let gridPatternInput1 =

`7283455864

6731158619

8988242643

3830589324

2229505813

5633845374

6473530293

7053106601

0834282956

4607924137`;

/\* Pattern Info which needs to search in above Grid. \*\*/

let testCasesRows1 = 3;

let testCasesColumns1 = 4;

let testCaseInput1 =

`9505

3845

3530`;

console.log(searchPatternInGrid(gridPatternRows1,gridPatternColumns1,gridPatternInput1,testCasesRows1,testCasesColumns1,testCaseInput1));

/\* Test-Case 2 - Grid Information like rows, columns. \*\*/

let gridPatternRows2 = 15;

let gridPatternColumns2 = 15;

let gridPatternInput2 =

`400453592126560

114213133098692

474386082879648

522356951189169

887109450487496

252802633388782

502771484966748

075975207693780

511799789562806

404007454272504

549043809916080

962410809534811

445893523733475

768705303214174

650629270887160`;

/\* Pattern Info which needs to search in above Grid. \*\*/

let testCasesRows2 = 2;

let testCasesColumns2 = 2;

let testCaseInput2 =

`99

99`;

console.log(searchPatternInGrid(gridPatternRows2,gridPatternColumns2,gridPatternInput2,testCasesRows2,testCasesColumns2,testCaseInput2));

/\* Test-Case 3 - Grid Information like rows, columns. \*\*/

let gridPatternRows3 = 20;

let gridPatternColumns3 = 20;

let gridPatternInput3 =

`34889246430321978567

58957542800420926643

35502505614464308821

14858224623252492823

72509980920257761017

22842014894387119401

01112950562348692493

16417403478999610594

79426411112116726706

65175742483779283052

89078730337964397201

13765228547239925167

26113704444636815161

25993216162800952044

88796416233981756034

14416627212117283516

15248825304941012863

88460496662793369385

59727291023618867708

19755940017808628326`;

/\* Pattern Info which needs to search in above Grid. \*\*/

let testCasesRows3 = 7;

let testCasesColumns3 = 4;

let testCaseInput3 =

`1641

7942

6517

8907

1376

2691

2599`;

console.log(searchPatternInGrid(gridPatternRows3,gridPatternColumns3,gridPatternInput3,testCasesRows3,testCasesColumns3,testCaseInput3));

/\* Test-Case 4 - Grid Information like rows, columns. \*\*/

let gridPatternRows4 = 25;

let gridPatternColumns4 = 25;

let gridPatternInput4 =

`7652157548860692421022503

9283597467877865303553675

4160389485250089289309493

2583470721457150497569300

3220130778636571709490905

3588873017660047694725749

9288991387848870159567061

4840101673383478700737237

8430916536880190158229898

8986106490042260460547150

2591460395957631878779378

1816190871689680423501920

0704047294563387014281341

8544774664056811258209321

9609294756392563447060526

0170173859593369054590795

6088985673796975810221577

7738800757919472437622349

5474120045253009653348388

3930491401877849249410013

1486477041403746396925337

2955579022827592919878713

2625547961868100985291514

3673299809851325174555652

4533398973801647859680907`;

/\* Pattern Info which needs to search in above Grid. \*\*/

let testCasesRows4 = 5;

let testCasesColumns4 = 4;

let testCaseInput4 =

`5250

1457

8636

7660

7848`;

console.log(searchPatternInGrid(gridPatternRows4,gridPatternColumns4,gridPatternInput4,testCasesRows4,testCasesColumns4,testCaseInput4));