**CS 700-34 Kmeans Clustering**

**Student:** Shuhua Song **Due Date: Submission Data:** Soft Copy: 04/05/2020 Soft Copy: 04/05/2020 Hard Copy: 04/05/2020 Hard Copy: 04/05/2020

**Algorithm Steps:**

**I. LoadPointSet(inFile, pointSet)**

Step 0: index ← 0 Step 1: x, y ← read from inFile Step 2: pointSet[index].Xcoord ← (double) x

pointSet[index].Ycoord ← (double) y pointSet[index].label← 0 // meaning: no label yet pointSet[index].Distance ← 99999.00

Step 3: index ++ Step 4: repeat step 1 to step 3 until the end of inFile

**II. KMeansClustering(pointset, K, displayAry, outFile)**

Step 0: iteration ← 0

choosedIndex //1D array with size K initial -1, for storing

the centroid index that already choosed

Step 1: iniDisplay(displayAry): initial the whole displayAry with 0

Step 2: selectKcentroids (pointSet, K, KcentroidAry)

Step 3: change← 4

Step 4: index ← 0

iteration++ change = 0

Step 5: pt ← pointSet[index] // get next pt, which is a Point class type!!

minDist ← pointSet[index]’s distance

Step 6: minLable ← DistanceMinLable (pt, KcentroidAry, minDist)

Step 7: if pointSet[index]’s label != minLabel

pointSet[index]’s label ← minLabel changes ++

Step 8: index ++

Step 9: repeat step5 — step8 unitil index >= numPts

Step 10: PlotDisplayAry (pointSet, displayAry)

PrettyPrint (displayAry, outFile1, iteration) computeCentroids (pointSet, KcentroidAry) Step 11: repeat step 4 to step 10 until change <= 2

**III. selectKcentroids(pointset, K, choosedIndex)**

Step 0: Kcnt ← 0

repeatYN ← true index = 0 Step 1: index ← randomly select an index from 0 to numPts-1 // Call a random generator function for this Step 2: repeatYN ← checkRepeat (index,choosedIndex) Step 3: repeat step 1 to step 2 until repeatYN is false Step 4: choosedIndex[Kcnt] ← index Step 5: KCnt ++

KcentroidAry [Kcnt].Xcoord ← pointSet[index].Xcoord KcentroidAry [Kcnt].Ycoord ← pointSet[index].Ycoord KcentroidAry [Kcnt].label← Kcnt // its own label KcentroidAry [Kcnt].Distance ← 0.0

Step 6: repeat step 1 to step 4 while Kcnt < K

**IV. checkRepeat(index, choosedIndex)**

Step 0: check if index is already exist in the array choosedInde Step 1: repeat step 0 until i >= K

**V. computeCentroids(pointset, KcentroidAry)**

Step 0: (double) sumX[] ← dynamically allocate 1-D array, size of K+1

// initialize to 0.0 (double) sumY[] ← dynamically allocate 1-D array, size of K+1,

// initialize to 0.0 (int) totalPt[] ← dynamically allocate 1-D array, size of K+1

// initialize to 0

Step 1: index ← 0

Step 2: label ← pointSet[index].label // get the point’s cluster label

sumX[label] += pointSet[index].Xcoord sumY[label] += pointSet[index].Ycoord totalPt[label] ++

Step 3: index++ Step 4: repeat step 2 to step 3 while index >= numPts step 5: label ← 1 step 6: if totalPt[label] > 0 KcentroidAry [label].Xcoord ←(sumX[label]/ totalPt[label]) KcentroidAry [label].Ycoord ←(sumY[label]/ totalPt[label])

Step 7: label ++ Step 8: repeat step 6 to step 7 while label <= K

**VI. DistanceMinLabel()**

Step 0: minDist ← 99999.00

minLabel ← 0 Step 1: label ← 1 Step 2: whichCentroid ← KcentroidAry[label] //whichCentroid is a Point class type Step 3: dist ← computeDist(pt, whichCentroid)// pt is a Point class type. Step 4: if dist < minDist

minLabel ← label minDist ←dist

Step 5: label ++

Pt.distance ← minDist Step 6: repeat step 2 to step 5 while label <= K Step 7: return minLabel

**VII. computeDist(point1, point2)**

Step 0: x1 ← point1.Xcoord y1← point1.Ycoord x2 ← point2.Xcoord y2← point2.Ycoord Step 1: return sqrt(pow(x2-x1, 2) + pow(y2-y1, 2) \* 1.0)

**VIII.** PlotDisplayAry(pointset, displayAry) Step 0: x ← (int) pointSet[i].Xcoord

y ← (int) pointset[i].Ycoord Step 1: displayAry[x][y] = pointset[i].label Step 2: repeat step 0 – step 1until i>= numPts

**IX.** prettyPrint() Step 0: if displayAry[i] > 0 → outFile displayAry[i][j]

Else outFile “ “ Step 1: repeat step 0 until i >= numRows && numCols

**Code:**

#include <iostream> #include <string> #include <fstream> #include <math.h> #include <vector> #include <random> #include <unordered\_set> using namespace std;

class Kmean { public:

class Point { public:

double Xcoord;//convert to int when plotting onto 2D displayAry double Ycoord; int label; double distance;

Point(){

Xcoord = 0; Ycoord = 0; label = 0; distance = 99999.00; //the distance to its own cluster centroid. } Point(double Xcoord, double Ycoord, int label, double distance){

this->Xcoord = Xcoord; this->Ycoord = Ycoord; this->label = label; this->distance = distance; } };

int K; // K clusters given in argv[2] int numPts; //The total number of points int numRows; int numCols; Point\* KcentroidAry;//No 0 as cluster label, cluster label from 1 to K

Kmean(){

int K = 0; int numPts = 0; int numRows = 0; int numCols = 0; }

Kmean(int K, int numPts, int numRows, int numCols){

this->K = K; this->numPts = numPts; this->numRows = numRows; this->numCols = numCols; KcentroidAry = new Point[K+1]; }

//Methods

void loadPointSet(ifstream& inFile, Point\* pointSet){

int index = 0; while(!inFile.eof()){

int x, y; inFile >> x; inFile >> y; pointSet[index].Xcoord = (double) x; pointSet[index].Ycoord = (double) y; pointSet[index].label = 0; //meaning there is no lable yet //pointSet[index].label = rand() % K + 1; pointSet[index].distance = 99999.00; index++; } }

void KMeansClustering(Point\* pointSet, int K, int\*\* displayAry, ofstream&

outFile){

int iteration = 0; int \*choosedIndex = new int[K]{-1}; //store the centroid index that already choosed

iniDisplay(displayAry); selectKcentroids(pointSet, K, choosedIndex); int change = 4;

while(change > 2){ int index = 0; iteration++; change = 0; while(index < numPts){

Point \*pt = &pointSet[index]; double minDist = pointSet[index].distance;//pt's distance int minLabel = distanceMinLable(pt, KcentroidAry,minDist); if(pointSet[index].label != minLabel){

pointSet[index].label = minLabel; change++; } index++; } plotDisplayAry(pointSet, displayAry); prettyPrint(displayAry, outFile, iteration); computeCentroids(pointSet, KcentroidAry); } } bool checkRepeat(int index, int\* choosedIndex){ //returns true is the index has been generated prior

for(int i=0; i<K; i++){

if(choosedIndex[i] == index){

return true;//the index has been generated } } return false; } void selectKcentroids(Point\* pointSet, int K, int\* choosedIndex) {

int Kcnt = 0; bool repeatYN = true; int index = 0;

while(Kcnt < K) {//Kcentroid's size is K+1, so we don't need Kcnt<=K

index = rand() % numPts; while(repeatYN){

repeatYN = checkRepeat(index, choosedIndex); index = rand() % numPts; } choosedIndex[Kcnt] = index; Kcnt++; KcentroidAry[Kcnt].Xcoord = pointSet[index].Xcoord; KcentroidAry[Kcnt].Ycoord = pointSet[index].Ycoord; KcentroidAry[Kcnt].label = Kcnt; KcentroidAry[Kcnt].distance = 0.0; } }

void iniDisplay(int\*\* displayAry){

for(int i=0; i<numRows; i++){

for(int j=0; j<numCols; j++){

displayAry[i][j] = 0;

} } }

int distanceMinLable(Point\* pt, Point\* KcentroidAry, double minDist) {

minDist = 99999.00; int minLabel = 0; int label = 1; while(label<=K){

Point\* whichCentroid = &KcentroidAry[label]; double dist = computeDist(pt, whichCentroid); if(dist < minDist){

minLabel = label; minDist = dist; } label++; pt->distance = minDist; } return minLabel; } int computeCentroids(Point\* pointSet, Point\* KcentroidAry){

double \*sumX = new double[K+1]{0.0}; double \*sumY = new double[K+1]{0.0}; int \*totalPt = new int[K+1]{0}; int index = 0; while(index < numPts){

int label = pointSet[index].label; sumX[label] += pointSet[index].Xcoord; sumY[label] += pointSet[index].Ycoord; totalPt[label]++; index++; } int label = 1; while(label <= K){

if(totalPt[label]>0){

KcentroidAry[label].Xcoord = sumX[label]/totalPt[label]; KcentroidAry[label].Ycoord = sumY[label]/totalPt[label]; label++; } } } double computeDist(Point\* pt, Point\* whichCentroid){

double x1 = pt->Xcoord; double y1 = pt->Ycoord; double x2 = whichCentroid->Xcoord; double y2 = whichCentroid->Ycoord; return sqrt(pow(x2-x1,2)+ pow(y2-y1,2) \* 1.0); }

void plotDisplayAry(Point\* pointSet, int\*\* displayAry){

for(int i=0; i<numPts; i++){

int x = (int) pointSet[i].Xcoord; int y = (int) pointSet[i].Ycoord; displayAry[x][y] = pointSet[i].label; } } void prettyPrint(int\*\* displayAry, ofstream& outFile, int iteration){

outFile << " K : " << K << " iteration :" << iteration; for(int i=0; i<numRows; i++){

for(int j=0; j<numCols; j++){ if(displayAry[i][j] > 0){

outFile << displayAry[i][j]; }else{

outFile << " "; } } outFile << endl; } } void printResult(Point\* pointSet, ofstream& outFile){

outFile << numRows << " " << numCols << endl; outFile << numPts << endl; int i = 0; while(i<numPts){

outFile << pointSet[i].Xcoord << " " << pointSet[i].Ycoord << " " << pointSet[i].label << endl; i++; } }

}; int main(int argc, char \*argv[]) { srand((unsigned)time(0));

ifstream inFile1(argv[1]); int K = stoi(argv[2]);

ofstream outFile1(argv[3]); ofstream outFile2(argv[4]);

int numRows, numCols, numPts; inFile1 >> numRows; inFile1 >> numCols; inFile1 >> numPts; cout << "numRows:" << numRows << " numCols:" << numCols << endl; cout << numPts << endl;

Kmean::Point \*pointSet = new Kmean::Point[numPts]; //1D array of Point class of size numPts

int\*\* displayAry = new int\*[numRows]; //a 2D array, size of numRows by numCols for displaying purposes

for(int i=0; i<numRows; i++){

displayAry[i] = new int[numCols]; }

int change = 0; Kmean \*kmean = new Kmean(K, numPts, numRows, numCols);

kmean->loadPointSet(inFile1, pointSet); kmean->KMeansClustering(pointSet, K, displayAry , outFile1);

kmean->printResult(pointSet, outFile2); inFile1.close(); outFile1.close(); outFile2.close(); return 0; }

**Print outFile1 and outFile2 for K = 4(with proper caption) OutFile1** K : 4 iteration :1

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 31

2 2233 3 33 3 333111 22 2 33 3 3 3 33 33 11 11

33 3 3 3 33 3 33111 33 33 33 3 3 3 3 111

33 33 3 3 3 3 11

3 333333 11 111 111 33 33 3 3 33 31 111

3 3 33 11 111 3 3 3 1111 111

1 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 44 44 4444 44 11 1 4444 4 44 44 411 1 444 44 44 44 44 11 44 44 44 44 41 444 444 44 44 1

444 444 4 1 444 44 44 1 1

444 444 444 411 1111

4 44 44 1 1111 4 444 411 1 11111 4 444444 411 111 1111

4 1 1111 44 444 1 1 1 1111 444 444 1 1111 44 44 1 1111

K : 4 iteration :2

22 2 2 22222 222 2

2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333 22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 44 44 4444 44 41 1 4444 4 44 44 444 1 444 44 44 44 44 41 44 44 44 44 44 444 444 44 44 4

444 444 4 1 444 44 44 4 1

444 444 444 441 1111

4 44 44 1 1111 4 444 444 1 11111 4 444444 444 111 1111

4 1 1111 44 444 1 1 1 1111 444 444 1 1111 44 44 1 1111

K : 4 iteration :3

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333

22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 224 44 111

4 4444

4 44 4444 44 44 4444 44 44 1 4444 4 44 44 444 4 444 44 44 44 44 44 44 44 44 44 44 444 444 44 44 4

444 444 4 4 444 44 44 4 1

444 444 444 444 1111

4 44 44 1 1111 4 444 444 1 11111 4 444444 444 111 1111

4 1 1111 44 444 1 1 1 1111 444 444 1 1111 44 44 1 1111

K : 4 iteration :4

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333 22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111

1 111111 111

1111 111 111

1111

1111 1111

111 222 44 111

2 4444

4 44 4444 44 44 4444 44 44 4 4444 4 44 44 444 4 444 44 44 44 44 44 44 44 44 44 44 444 444 44 44 4

444 444 4 4 444 44 44 4 1

444 444 444 444 1111

4 44 44 1 1111 4 444 444 1 11111 4 444444 444 111 1111

4 1 1111 44 444 1 1 1 1111 444 444 1 1111 44 44 1 1111

K : 4 iteration :5

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333 22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111

222 44 111

2 4444

4 44 4444 44 44 4444 44 44 4 4444 4 44 44 444 4 444 44 44 44 44 44 44 44 44 44 44 444 444 44 44 4

444 444 4 4 444 44 44 4 1

444 444 444 444 1111

4 44 44 1 1111 4 444 444 1 11111 4 444444 444 111 1111

4 1 1111 44 444 1 1 1 1111 444 444 1 1111 44 44 1 1111 **outFile2** 45 60 399 1 6 2 1 7 2 2 4 2 3 3 2 3 5 2 3 6 2 3 7 2 3 8 2 3 9 2 4 4 2 4 5 2 4 6 2 4 8 2 5 3 2 5 5 2 5 7 2 6 2 2 6 3 2 6 4 2 6 5 2 6 8 2 6 9 2 6 10 2 6 25 3 6 26 3 6 28 3 6 29 3 7 2 2 7 4 2 7 5 2 7 6 2

7 7 2 7 8 2 7 9 2 7 10 2 7 24 3 7 26 3 7 27 3 7 28 3 7 29 3 7 30 3 8 2 2 8 3 2 8 4 2 8 5 2 8 7 2 8 20 3 8 25 3 8 27 3 8 28 3 8 29 3 8 31 3 8 32 3 9 3 2 9 6 2 9 7 2 9 8 2 9 9 2 9 21 3 9 24 3 9 25 3 9 27 3 9 29 3 9 30 3 9 31 3 9 32 3 9 33 3 9 34 3 10 2 2 10 3 2 10 5 2 10 7 2 10 8 2 10 10 2 10 22 3 10 25 3 10 27 3 10 28 3 10 30 3 10 31 3 10 33 3 10 34 3 10 36 3

10 37 3 11 4 2 11 5 2 11 7 2 11 9 2 11 12 2 11 22 3 11 23 3 11 27 3 11 30 3 11 31 3 11 32 3 11 33 3 11 34 3 12 3 2 12 4 2 12 6 2 12 7 2 12 10 2 12 11 2 12 25 3 12 27 3 12 29 3 12 31 3 12 51 1 12 52 1 12 53 1 13 5 2 13 6 2 13 8 2 13 9 2 13 24 3 13 26 3 13 28 3 13 30 3 13 32 3 13 33 3 14 7 2 14 25 3 14 26 3 14 27 3 14 28 3 14 29 3 14 30 3 14 32 3 14 33 3 14 49 1 14 50 1 14 51 1 14 54 1 14 55 1 14 56 1

15 4 2 15 5 2 15 9 2 15 10 2 15 24 3 15 26 3 15 28 3 15 29 3 15 31 3 15 32 3 15 51 1 15 52 1 15 53 1 16 23 3 16 27 3 16 29 3 16 30 3 16 32 3 16 33 3 16 52 1 16 53 1 16 54 1 17 22 3 17 26 3 17 28 3 17 50 1 17 51 1 17 52 1 17 53 1 17 55 1 17 56 1 17 57 1 18 31 3 18 49 1 18 53 1 18 54 1 18 55 1 18 56 1 19 45 1 19 47 1 19 48 1 19 49 1 19 50 1 19 51 1 19 52 1 19 54 1 19 55 1 19 56 1 20 51 1 20 52 1 20 53 1 20 54 1

21 48 1 21 49 1 21 50 1 21 55 1 21 56 1 21 57 1 22 50 1 22 51 1 22 52 1 22 53 1 23 53 1 23 54 1 23 55 1 23 56 1 24 50 1 24 51 1 24 52 1 24 53 1 25 55 1 25 56 1 25 57 1 26 2 2 26 3 2 26 4 2 26 7 4 26 8 4 26 52 1 26 53 1 26 54 1 27 3 2 27 5 4 27 6 4 27 7 4 27 8 4 28 4 4 28 6 4 28 7 4 28 9 4 28 10 4 28 11 4 28 12 4 28 29 4 28 30 4 29 2 4 29 3 4 29 5 4 29 6 4 29 7 4 29 8 4 29 29 4 29 30 4 29 32 4

29 33 4 29 36 4 30 2 4 30 3 4 30 4 4 30 5 4 30 7 4 30 10 4 30 11 4 30 26 4 30 27 4 30 31 4 30 32 4 30 33 4 30 35 4 31 1 4 31 2 4 31 3 4 31 5 4 31 6 4 31 9 4 31 10 4 31 26 4 31 27 4 31 29 4 31 30 4 31 33 4 31 34 4 32 2 4 32 3 4 32 6 4 32 7 4 32 25 4 32 26 4 32 28 4 32 29 4 32 32 4 32 33 4 33 3 4 33 4 4 33 5 4 33 7 4 33 8 4 33 9 4 33 26 4 33 27 4 33 30 4 33 31 4 33 33 4 34 5 4 34 6 4 34 7 4

34 28 4 34 29 4 34 30 4 34 32 4 34 35 4 35 1 4 35 2 4 35 3 4 35 26 4 35 27 4 35 30 4 35 31 4 35 33 4 35 53 1 36 3 4 36 4 4 36 5 4 36 25 4 36 26 4 36 27 4 36 29 4 36 30 4 36 31 4 36 33 4 36 34 4 36 35 4 36 52 1 36 53 1 36 54 1 36 55 1 37 24 4 37 26 4 37 27 4 37 30 4 37 31 4 37 50 1 37 53 1 37 54 1 37 55 1 37 56 1 38 23 4 38 29 4 38 30 4 38 31 4 38 33 4 38 34 4 38 35 4 38 48 1 38 51 1 38 52 1 38 53 1 38 54 1

38 55 1 39 21 4 39 26 4 39 27 4 39 28 4 39 29 4 39 30 4 39 31 4 39 33 4 39 34 4 39 35 4 39 49 1 39 50 1 39 51 1 39 54 1 39 55 1 39 56 1 39 57 1 40 32 4 40 48 1 40 51 1 40 52 1 40 53 1 40 54 1 41 26 4 41 27 4 41 29 4 41 30 4 41 31 4 41 47 1 41 49 1 41 52 1 41 55 1 41 56 1 41 57 1 41 58 1 42 28 4 42 29 4 42 30 4 42 32 4 42 33 4 42 34 4 42 50 1 42 53 1 42 54 1 42 55 1 42 56 1 43 26 4 43 27 4 43 30 4 43 31 4 43 49 1

43 52 1 43 53 1 43 54 1 43 55 1

**Print outFile1 and outFile2 for K = 5(with proper caption) outFile1**

K : 5 iteration :1

22 2 2 22223 222 2 2 2 2 2222 333 11 11 2 2223333 1 11111 2222 3 1 1 111 11

2 3333 1 11 1 111111 22 2 33 3 1 1 11 11 11 11

23 3 3 3 55 1 11111 22 33 33 5 1 1 1 111

33 33 5 5 1 1 11

3 555511 11 111 111 33 33 5 5 55 11 111

5 5 55 11 111 5 5 5 1111 111

5 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 455 555 555 5555

4 55 55 5 5555 4 555 555 5 55555 4 445555 555 555 5555

5 5 5555 44 555 5 5 5 5555 445 555 5 5555

44 45 5 5555

K : 5 iteration :2

22 2 2 22222 222 2 2 2 2 2222 233 31 11 2 2222333 3 11111 2222 3 3 3 111 11

2 3333 3 33 1 111111 22 3 33 3 3 3 11 11 11 11

33 3 3 3 33 1 11111 33 33 33 1 1 1 1 111

33 33 3 1 1 1 11

3 111111 11 111 111 33 33 3 1 11 11 111

3 1 11 11 111 3 1 1 1111 111

1 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 555 555 555 5555

5 55 55 5 5555 4 555 555 5 55555 4 555555 555 555 5555

5 5 5555 55 555 5 5 5 5555 555 555 5 5555 55 55 5 5555

K : 5 iteration :3

22 2 2 22222 222 2

2 2 2 2222 222 33 11 2 2222223 3 33111 2222 2 3 3 311 11

2 2233 3 33 3 111111 22 2 23 3 3 3 31 11 11 11

22 3 3 3 33 3 11111 22 33 33 3 1 1 1 111

23 33 3 3 1 1 11

3 331111 11 111 111 33 33 3 3 11 11 111

3 1 11 11 111 3 3 1 1111 111

1 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 555 555 555 5555

5 55 55 5 5555 5 555 555 5 55555 4 555555 555 555 5555

5 5 5555 55 555 5 5 5 5555 555 555 5 5555 55 55 5 5555

K : 5 iteration :4

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 31

2 2222 3 33 3 333111 22 2 22 2 3 3 33 33 11 11

22 2 2 3 33 3 31111

22 22 23 3 3 3 1 111

22 22 3 3 3 3 11

2 333333 11 111 111 22 23 3 3 33 11 111

3 3 31 11 111 3 3 3 1111 111

1 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 555 555 555 5555

5 55 55 5 5555 5 555 555 5 55555 5 555555 555 555 5555

5 5 5555 55 555 5 5 5 5555 555 555 5 5555 55 55 5 5555

K : 5 iteration :5

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333 22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111

1 111111 111

1111 111 111

1111

1111 1111

111 444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 555 555 555 5555

5 55 55 5 5555 5 555 555 5 55555 5 555555 555 555 5555

5 5 5555 55 555 5 5 5 5555 555 555 5 5555 55 55 5 5555

K : 5 iteration :6

22 2 2 22222 222 2 2 2 2 2222 222 33 33 2 2222222 3 33333 2222 2 3 3 333 33

2 2222 3 33 3 333333 22 2 22 2 3 3 33 33 33 33

22 2 2 2 33 3 33333 22 22 22 3 3 3 3 111

22 22 3 3 3 3 33

2 333333 33 111 111 22 22 3 3 33 33 111

3 3 33 33 111 3 3 3 1111 111

3 1 1111 1 111111 111

1111 111 111

1111

1111 1111

111

444 44 111

4 4444

4 44 4444 55 44 4444 55 55 5 4444 4 44 55 555 5 444 44 44 55 55 55 44 44 55 55 55 444 444 55 55 5

444 555 5 5 444 55 55 5 5

444 555 555 555 5551

5 55 55 5 5551 5 555 555 5 55555 5 555555 555 555 5555

5 5 5555 55 555 5 5 5 5555 555 555 5 5555 55 55 5 5555

**OutFile2**

45 60 399 1 6 2 1 7 2 2 4 2 3 3 2 3 5 2 3 6 2 3 7 2 3 8 2 3 9 2 4 4 2 4 5 2 4 6 2 4 8 2 5 3 2 5 5 2 5 7 2 6 2 2 6 3 2 6 4 2 6 5 2 6 8 2 6 9 2 6 10 2 6 25 3 6 26 3 6 28 3 6 29 3 7 2 2 7 4 2

7 5 2 7 6 2 7 7 2 7 8 2 7 9 2 7 10 2 7 24 3 7 26 3 7 27 3 7 28 3 7 29 3 7 30 3 8 2 2 8 3 2 8 4 2 8 5 2 8 7 2 8 20 3 8 25 3 8 27 3 8 28 3 8 29 3 8 31 3 8 32 3 9 3 2 9 6 2 9 7 2 9 8 2 9 9 2 9 21 3 9 24 3 9 25 3 9 27 3 9 29 3 9 30 3 9 31 3 9 32 3 9 33 3 9 34 3 10 2 2 10 3 2 10 5 2 10 7 2 10 8 2 10 10 2 10 22 3 10 25 3 10 27 3 10 28 3 10 30 3 10 31 3 10 33 3

10 34 3 10 36 3 10 37 3 11 4 2 11 5 2 11 7 2 11 9 2 11 12 2 11 22 3 11 23 3 11 27 3 11 30 3 11 31 3 11 32 3 11 33 3 11 34 3 12 3 2 12 4 2 12 6 2 12 7 2 12 10 2 12 11 2 12 25 3 12 27 3 12 29 3 12 31 3 12 51 1 12 52 1 12 53 1 13 5 2 13 6 2 13 8 2 13 9 2 13 24 3 13 26 3 13 28 3 13 30 3 13 32 3 13 33 3 14 7 2 14 25 3 14 26 3 14 27 3 14 28 3 14 29 3 14 30 3 14 32 3 14 33 3 14 49 1 14 50 1 14 51 1 14 54 1

14 55 1 14 56 1 15 4 2 15 5 2 15 9 2 15 10 2 15 24 3 15 26 3 15 28 3 15 29 3 15 31 3 15 32 3 15 51 1 15 52 1 15 53 1 16 23 3 16 27 3 16 29 3 16 30 3 16 32 3 16 33 3 16 52 1 16 53 1 16 54 1 17 22 3 17 26 3 17 28 3 17 50 1 17 51 1 17 52 1 17 53 1 17 55 1 17 56 1 17 57 1 18 31 3 18 49 1 18 53 1 18 54 1 18 55 1 18 56 1 19 45 1 19 47 1 19 48 1 19 49 1 19 50 1 19 51 1 19 52 1 19 54 1 19 55 1 19 56 1 20 51 1 20 52 1

20 53 1 20 54 1 21 48 1 21 49 1 21 50 1 21 55 1 21 56 1 21 57 1 22 50 1 22 51 1 22 52 1 22 53 1 23 53 1 23 54 1 23 55 1 23 56 1 24 50 1 24 51 1 24 52 1 24 53 1 25 55 1 25 56 1 25 57 1 26 2 4 26 3 4 26 4 4 26 7 4 26 8 4 26 52 1 26 53 1 26 54 1 27 3 4 27 5 4 27 6 4 27 7 4 27 8 4 28 4 4 28 6 4 28 7 4 28 9 4 28 10 4 28 11 4 28 12 4 28 29 5 28 30 5 29 2 4 29 3 4 29 5 4 29 6 4 29 7 4 29 8 4 29 29 5

29 30 5 29 32 5 29 33 5 29 36 5 30 2 4 30 3 4 30 4 4 30 5 4 30 7 4 30 10 4 30 11 4 30 26 5 30 27 5 30 31 5 30 32 5 30 33 5 30 35 5 31 1 4 31 2 4 31 3 4 31 5 4 31 6 4 31 9 4 31 10 4 31 26 5 31 27 5 31 29 5 31 30 5 31 33 5 31 34 5 32 2 4 32 3 4 32 6 4 32 7 4 32 25 5 32 26 5 32 28 5 32 29 5 32 32 5 32 33 5 33 3 4 33 4 4 33 5 4 33 7 4 33 8 4 33 9 4 33 26 5 33 27 5 33 30 5 33 31 5 33 33 5 34 5 4

34 6 4 34 7 4 34 28 5 34 29 5 34 30 5 34 32 5 34 35 5 35 1 4 35 2 4 35 3 4 35 26 5 35 27 5 35 30 5 35 31 5 35 33 5 35 53 5 36 3 4 36 4 4 36 5 4 36 25 5 36 26 5 36 27 5 36 29 5 36 30 5 36 31 5 36 33 5 36 34 5 36 35 5 36 52 5 36 53 5 36 54 5 36 55 1 37 24 5 37 26 5 37 27 5 37 30 5 37 31 5 37 50 5 37 53 5 37 54 5 37 55 5 37 56 1 38 23 5 38 29 5 38 30 5 38 31 5 38 33 5 38 34 5 38 35 5 38 48 5 38 51 5 38 52 5

38 53 5 38 54 5 38 55 5 39 21 5 39 26 5 39 27 5 39 28 5 39 29 5 39 30 5 39 31 5 39 33 5 39 34 5 39 35 5 39 49 5 39 50 5 39 51 5 39 54 5 39 55 5 39 56 5 39 57 5 40 32 5 40 48 5 40 51 5 40 52 5 40 53 5 40 54 5 41 26 5 41 27 5 41 29 5 41 30 5 41 31 5 41 47 5 41 49 5 41 52 5 41 55 5 41 56 5 41 57 5 41 58 5 42 28 5 42 29 5 42 30 5 42 32 5 42 33 5 42 34 5 42 50 5 42 53 5 42 54 5 42 55 5 42 56 5 43 26 5 43 27 5 43 30 5

43 31 5 43 49 5 43 52 5 43 53 5 43 54 5 43 55 5

**Print outFile1 and outFile2 for K = 6(with proper caption) outFile1** K : 6 iteration :1

66 6 6 66666 666 6 6 6 6 6666 666 66 22 6 6666666 6 62222 6666 6 6 6 222 33

6 6666 6 66 2 223333 66 6 66 6 6 6 22 33 33 33

66 6 6 6 66 2 33333 66 66 66 6 2 3 3 333

66 66 6 6 2 3 33

6 662333 33 333 333 11 66 6 6 33 33 333

6 3 33 33 333 6 6 3 3333 333

3 3 3333 3 333333 355

3335 333 555

3555

5555 5555

555 111 11 555

1 1111

1 11 1111 44 11 1111 44 44 4 1111 1 11 44 444 4 111 11 11 44 44 44 11 11 44 44 44 111 111 44 44 4

111 444 4 4 111 44 44 4 5

111 444 444 444 5555

4 44 44 5 5555 4 444 444 5 55555 4 444444 444 555 5555

4 5 5555 44 444 5 5 5 5555 444 444 5 5555

44 44 5 5555

K : 6 iteration :2

66 6 6 66666 666 6 6 6 6 6666 666 22 22 6 6666666 2 22222 6666 6 2 2 222 22

6 6666 2 22 2 222222 66 6 66 6 2 2 22 22 22 33

66 6 6 6 22 2 22222 66 66 66 2 2 2 2 333

66 66 2 2 2 2 22

6 222222 22 333 333 66 66 2 2 22 22 333

2 2 22 22 333 2 2 2 3333 333

2 3 3333 3 333333 333

3333 333 555

3355

5555 5555

555 111 11 555

1 1111

1 11 1111 44 11 1111 44 44 4 1111 1 11 44 444 4 111 11 11 44 44 44 11 11 44 44 44 111 111 44 44 4

111 444 4 4 111 44 44 4 5

111 444 444 444 5555

4 44 44 5 5555 4 444 444 5 55555 4 444444 444 555 5555

4 5 5555 44 444 5 5 5 5555 444 444 5 5555 44 44 5 5555

K : 6 iteration :3

66 6 6 66666 666 6

6 6 6 6666 666 22 22 6 6666666 2 22222 6666 6 2 2 222 22

6 6666 2 22 2 222222 66 6 66 6 2 2 22 22 22 22

66 6 6 6 22 2 22222 66 66 66 2 2 2 2 333

66 66 2 2 2 2 22

6 222222 22 333 333 66 66 2 2 22 22 333

2 2 22 22 333 2 2 2 3333 333

2 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 111 11 555

1 1111

1 11 1111 44 11 1111 44 44 4 1111 1 11 44 444 4 111 11 11 44 44 44 11 11 44 44 44 111 111 44 44 4

111 444 4 4 111 44 44 4 5

111 444 444 444 5555

4 44 44 5 5555 4 444 444 5 55555 4 444444 444 555 5555

4 5 5555 44 444 5 5 5 5555 444 444 5 5555 44 44 5 5555

K : 6 iteration :4

66 6 6 66666 666 6 6 6 6 6666 666 22 22 6 6666666 2 22222 6666 6 2 2 222 22

6 6666 2 22 2 222222 66 6 66 6 2 2 22 22 22 22

66 6 6 6 22 2 22222

66 66 66 2 2 2 2 333

66 66 2 2 2 2 22

6 222222 22 333 333 66 66 2 2 22 22 333

2 2 22 22 333 2 2 2 3333 333

2 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 111 11 333

1 1111

1 11 1111 44 11 1111 44 44 4 1111 1 11 44 444 4 111 11 11 44 44 44 11 11 44 44 44 111 111 44 44 4

111 444 4 4 111 44 44 4 5

111 444 444 444 5555

4 44 44 5 5555 4 444 444 5 55555 4 444444 444 555 5555

4 5 5555 44 444 5 5 5 5555 444 444 5 5555 44 44 5 5555

K : 6 iteration :5

66 6 6 66666 666 6 6 6 6 6666 666 22 22 6 6666666 2 22222 6666 6 2 2 222 22

6 6666 2 22 2 222222 66 6 66 6 2 2 22 22 22 22

66 6 6 6 22 2 22222 66 66 66 2 2 2 2 333

66 66 2 2 2 2 22

6 222222 22 333 333 66 66 2 2 22 22 333

2 2 22 22 333 2 2 2 3333 333

2 3 3333

3 333333 333

3333 333 333

3333

3333 3333

333 111 11 333

1 1111

1 11 1111 44 11 1111 44 44 4 1111 1 11 44 444 4 111 11 11 44 44 44 11 11 44 44 44 111 111 44 44 4

111 444 4 4 111 44 44 4 5

111 444 444 444 5555

4 44 44 5 5555 4 444 444 5 55555 4 444444 444 555 5555

4 5 5555 44 444 5 5 5 5555 444 444 5 5555 44 44 5 5555

**OutFile2**

45 60 399 1 6 6 1 7 6 2 4 6 3 3 6 3 5 6 3 6 6 3 7 6 3 8 6 3 9 6 4 4 6 4 5 6 4 6 6 4 8 6 5 3 6 5 5 6 5 7 6 6 2 6 6 3 6 6 4 6 6 5 6 6 8 6

6 9 6 6 10 6 6 25 2 6 26 2 6 28 2 6 29 2 7 2 6 7 4 6 7 5 6 7 6 6 7 7 6 7 8 6 7 9 6 7 10 6 7 24 2 7 26 2 7 27 2 7 28 2 7 29 2 7 30 2 8 2 6 8 3 6 8 4 6 8 5 6 8 7 6 8 20 2 8 25 2 8 27 2 8 28 2 8 29 2 8 31 2 8 32 2 9 3 6 9 6 6 9 7 6 9 8 6 9 9 6 9 21 2 9 24 2 9 25 2 9 27 2 9 29 2 9 30 2 9 31 2 9 32 2 9 33 2 9 34 2 10 2 6 10 3 6 10 5 6 10 7 6 10 8 6

10 10 6 10 22 2 10 25 2 10 27 2 10 28 2 10 30 2 10 31 2 10 33 2 10 34 2 10 36 2 10 37 2 11 4 6 11 5 6 11 7 6 11 9 6 11 12 6 11 22 2 11 23 2 11 27 2 11 30 2 11 31 2 11 32 2 11 33 2 11 34 2 12 3 6 12 4 6 12 6 6 12 7 6 12 10 6 12 11 6 12 25 2 12 27 2 12 29 2 12 31 2 12 51 3 12 52 3 12 53 3 13 5 6 13 6 6 13 8 6 13 9 6 13 24 2 13 26 2 13 28 2 13 30 2 13 32 2 13 33 2 14 7 6 14 25 2 14 26 2 14 27 2 14 28 2

14 29 2 14 30 2 14 32 2 14 33 2 14 49 3 14 50 3 14 51 3 14 54 3 14 55 3 14 56 3 15 4 6 15 5 6 15 9 6 15 10 6 15 24 2 15 26 2 15 28 2 15 29 2 15 31 2 15 32 2 15 51 3 15 52 3 15 53 3 16 23 2 16 27 2 16 29 2 16 30 2 16 32 2 16 33 2 16 52 3 16 53 3 16 54 3 17 22 2 17 26 2 17 28 2 17 50 3 17 51 3 17 52 3 17 53 3 17 55 3 17 56 3 17 57 3 18 31 2 18 49 3 18 53 3 18 54 3 18 55 3 18 56 3 19 45 3 19 47 3 19 48 3 19 49 3

19 50 3 19 51 3 19 52 3 19 54 3 19 55 3 19 56 3 20 51 3 20 52 3 20 53 3 20 54 3 21 48 3 21 49 3 21 50 3 21 55 3 21 56 3 21 57 3 22 50 3 22 51 3 22 52 3 22 53 3 23 53 3 23 54 3 23 55 3 23 56 3 24 50 3 24 51 3 24 52 3 24 53 3 25 55 3 25 56 3 25 57 3 26 2 1 26 3 1 26 4 1 26 7 1 26 8 1 26 52 3 26 53 3 26 54 3 27 3 1 27 5 1 27 6 1 27 7 1 27 8 1 28 4 1 28 6 1 28 7 1 28 9 1 28 10 1 28 11 1 28 12 1 28 29 4

28 30 4 29 2 1 29 3 1 29 5 1 29 6 1 29 7 1 29 8 1 29 29 4 29 30 4 29 32 4 29 33 4 29 36 4 30 2 1 30 3 1 30 4 1 30 5 1 30 7 1 30 10 1 30 11 1 30 26 4 30 27 4 30 31 4 30 32 4 30 33 4 30 35 4 31 1 1 31 2 1 31 3 1 31 5 1 31 6 1 31 9 1 31 10 1 31 26 4 31 27 4 31 29 4 31 30 4 31 33 4 31 34 4 32 2 1 32 3 1 32 6 1 32 7 1 32 25 4 32 26 4 32 28 4 32 29 4 32 32 4 32 33 4 33 3 1 33 4 1 33 5 1 33 7 1

33 8 1 33 9 1 33 26 4 33 27 4 33 30 4 33 31 4 33 33 4 34 5 1 34 6 1 34 7 1 34 28 4 34 29 4 34 30 4 34 32 4 34 35 4 35 1 1 35 2 1 35 3 1 35 26 4 35 27 4 35 30 4 35 31 4 35 33 4 35 53 5 36 3 1 36 4 1 36 5 1 36 25 4 36 26 4 36 27 4 36 29 4 36 30 4 36 31 4 36 33 4 36 34 4 36 35 4 36 52 5 36 53 5 36 54 5 36 55 5 37 24 4 37 26 4 37 27 4 37 30 4 37 31 4 37 50 5 37 53 5 37 54 5 37 55 5 37 56 5 38 23 4 38 29 4

38 30 4 38 31 4 38 33 4 38 34 4 38 35 4 38 48 5 38 51 5 38 52 5 38 53 5 38 54 5 38 55 5 39 21 4 39 26 4 39 27 4 39 28 4 39 29 4 39 30 4 39 31 4 39 33 4 39 34 4 39 35 4 39 49 5 39 50 5 39 51 5 39 54 5 39 55 5 39 56 5 39 57 5 40 32 4 40 48 5 40 51 5 40 52 5 40 53 5 40 54 5 41 26 4 41 27 4 41 29 4 41 30 4 41 31 4 41 47 5 41 49 5 41 52 5 41 55 5 41 56 5 41 57 5 41 58 5 42 28 4 42 29 4 42 30 4 42 32 4 42 33 4 42 34 4

42 50 5 42 53 5 42 54 5 42 55 5 42 56 5 43 26 4 43 27 4 43 30 4 43 31 4 43 49 5 43 52 5 43 53 5 43 54 5 43 55 5

**Print outFile1 and outFile2 for K = 7(with proper caption) outFile1**

K : 7 iteration :1

44 4 4 44444 444 4 4 4 4 4444 444 44 41 2 4444444 4 44411 2244 4 4 4 411 11

2 4444 4 44 1 111111 22 2 44 4 4 4 11 11 11 11

22 4 4 4 44 1 11111 22 22 44 6 1 1 1 333

22 24 6 1 1 1 11

2 611111 11 333 333 22 22 6 6 11 11 333

6 1 11 11 333 6 6 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 11 1 7777 7 77 66 611 1 777 77 77 66 66 11 77 77 66 66 61

777 777 66 66 1

777 666 6 1 777 66 66 6 3

777 666 666 661 3333

6 66 66 3 3333 5 555 561 1 33333 5 555555 555 113 3333

5 1 3333 55 555 1 1 3 3333 555 555 1 3333 55 55 1 3333

K : 7 iteration :2

44 4 2 44444 244 4 2 2 4 2222 444 41 11 2 2224444 4 11111 2222 2 4 1 111 11

2 2224 4 41 1 111111 22 2 22 4 4 1 11 11 11 11

22 2 2 4 44 1 11111 22 22 24 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 5 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :3

44 2 2 24444 222 4 2 2 4 2222 444 11 11 2 2222444 1 11111 2222 2 4 1 111 11

2 2222 1 11 1 111111 22 2 22 4 1 1 11 11 11 11

22 2 2 4 11 1 11111 22 22 24 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :4

44

4 2 44444 244 4 2 2 4 2222 444 11 11 2 2224444 1 11111 2222 2 1 1 111 11

2 2224 1 11 1 111111 22 2 22 4 1 1 11 11 11 11

22 2 2 4 11 1 11111 22 22 22 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :5

44 4 4 44444 444 4 4 4 4 2244 444 11 11 2 2244444 1 11111 2222 2 1 1 111 11

2 2222 1 11 1 111111 22 2 22 2 1 1 11 11 11 11

22 2 2 2 11 1 11111 22 22 22 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :6

44 4 4 44444 444 4 4 4 4 4444 444 11 11 4 4444444 1 11111 2222 2 1 1 111 11

2 2222 1 11 1 111111 22 2 22 2 1 1 11 11 11 11

22 2 2 2 11 1 11111 22 22 22 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :7

44 4 4 44444 444 4 4 4 4 4444 444 11 11 4 4444444 1 11111 4444 2 1 1 111 11

2 2222 1 11 1 111111 22 2 22 2 1 1 11 11 11 11

22 2 2 2 11 1 11111 22 22 22 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6 7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

K : 7 iteration :8

44 4 4 44444 444 4 4 4 4 4444 444 11 11 4 4444444 1 11111 4444 4 1 1 111 11

2 2222 1 11 1 111111 22 2 22 2 1 1 11 11 11 11

22 2 2 2 11 1 11111 22 22 22 1 1 1 1 333

22 22 1 1 1 1 11

2 111111 11 333 333 22 22 1 1 11 11 333

1 1 11 11 333 1 1 1 3333 333

1 3 3333 3 333333 333

3333 333 333

3333

3333 3333

333 777 77 333

7 7777

7 77 7777 66 77 7777 66 66 6

7777 7 77 66 666 6 777 77 77 66 66 66 77 77 66 66 66 777 777 66 66 6

777 666 6 6 777 66 66 6 3

777 555 555 555 3333

5 55 55 3 3333 5 555 555 3 33333 5 555555 555 333 3333

5 3 3333 55 555 3 3 3 3333 555 555 3 3333 55 55 3 3333

**OutFile2**

45 60 399 1 6 4 1 7 4 2 4 4 3 3 4 3 5 4 3 6 4 3 7 4 3 8 4 3 9 4 4 4 4 4 5 4 4 6 4 4 8 4 5 3 4 5 5 4 5 7 4 6 2 4 6 3 4 6 4 4 6 5 4 6 8 4 6 9 4 6 10 4 6 25 1 6 26 1 6 28 1 6 29 1 7 2 4 7 4 4

7 5 4 7 6 4 7 7 4 7 8 4 7 9 4 7 10 4 7 24 1 7 26 1 7 27 1 7 28 1 7 29 1 7 30 1 8 2 4 8 3 4 8 4 4 8 5 4 8 7 4 8 20 1 8 25 1 8 27 1 8 28 1 8 29 1 8 31 1 8 32 1 9 3 2 9 6 2 9 7 2 9 8 2 9 9 2 9 21 1 9 24 1 9 25 1 9 27 1 9 29 1 9 30 1 9 31 1 9 32 1 9 33 1 9 34 1 10 2 2 10 3 2 10 5 2 10 7 2 10 8 2 10 10 2 10 22 1

10 25 1 10 27 1 10 28 1 10 30 1 10 31 1 10 33 1 10 34 1 10 36 1 10 37 1 11 4 2 11 5 2 11 7 2 11 9 2 11 12 2 11 22 1 11 23 1 11 27 1 11 30 1 11 31 1 11 32 1 11 33 1 11 34 1 12 3 2 12 4 2 12 6 2 12 7 2 12 10 2 12 11 2 12 25 1 12 27 1 12 29 1 12 31 1 12 51 3 12 52 3 12 53 3 13 5 2 13 6 2 13 8 2 13 9 2 13 24 1 13 26 1 13 28 1 13 30 1 13 32 1 13 33 1 14 7 2

14 25 1 14 26 1 14 27 1 14 28 1 14 29 1 14 30 1 14 32 1 14 33 1 14 49 3 14 50 3 14 51 3 14 54 3 14 55 3 14 56 3 15 4 2 15 5 2 15 9 2 15 10 2 15 24 1 15 26 1 15 28 1 15 29 1 15 31 1 15 32 1 15 51 3 15 52 3 15 53 3 16 23 1 16 27 1 16 29 1 16 30 1 16 32 1 16 33 1 16 52 3 16 53 3 16 54 3 17 22 1 17 26 1 17 28 1 17 50 3 17 51 3 17 52 3 17 53 3 17 55 3 17 56 3 17 57 3

18 31 1 18 49 3 18 53 3 18 54 3 18 55 3 18 56 3 19 45 3 19 47 3 19 48 3 19 49 3 19 50 3 19 51 3 19 52 3 19 54 3 19 55 3 19 56 3 20 51 3 20 52 3 20 53 3 20 54 3 21 48 3 21 49 3 21 50 3 21 55 3 21 56 3 21 57 3 22 50 3 22 51 3 22 52 3 22 53 3 23 53 3 23 54 3 23 55 3 23 56 3 24 50 3 24 51 3 24 52 3 24 53 3 25 55 3 25 56 3 25 57 3 26 2 7 26 3 7 26 4 7 26 7 7 26 8 7

26 52 3 26 53 3 26 54 3 27 3 7 27 5 7 27 6 7 27 7 7 27 8 7 28 4 7 28 6 7 28 7 7 28 9 7 28 10 7 28 11 7 28 12 7 28 29 6 28 30 6 29 2 7 29 3 7 29 5 7 29 6 7 29 7 7 29 8 7 29 29 6 29 30 6 29 32 6 29 33 6 29 36 6 30 2 7 30 3 7 30 4 7 30 5 7 30 7 7 30 10 7 30 11 7 30 26 6 30 27 6 30 31 6 30 32 6 30 33 6 30 35 6 31 1 7 31 2 7 31 3 7 31 5 7 31 6 7

31 9 7 31 10 7 31 26 6 31 27 6 31 29 6 31 30 6 31 33 6 31 34 6 32 2 7 32 3 7 32 6 7 32 7 7 32 25 6 32 26 6 32 28 6 32 29 6 32 32 6 32 33 6 33 3 7 33 4 7 33 5 7 33 7 7 33 8 7 33 9 7 33 26 6 33 27 6 33 30 6 33 31 6 33 33 6 34 5 7 34 6 7 34 7 7 34 28 6 34 29 6 34 30 6 34 32 6 34 35 6 35 1 7 35 2 7 35 3 7 35 26 6 35 27 6 35 30 6 35 31 6 35 33 6 35 53 3

36 3 7 36 4 7 36 5 7 36 25 5 36 26 5 36 27 5 36 29 5 36 30 5 36 31 5 36 33 5 36 34 5 36 35 5 36 52 3 36 53 3 36 54 3 36 55 3 37 24 5 37 26 5 37 27 5 37 30 5 37 31 5 37 50 3 37 53 3 37 54 3 37 55 3 37 56 3 38 23 5 38 29 5 38 30 5 38 31 5 38 33 5 38 34 5 38 35 5 38 48 3 38 51 3 38 52 3 38 53 3 38 54 3 38 55 3 39 21 5 39 26 5 39 27 5 39 28 5 39 29 5 39 30 5 39 31 5

39 33 5 39 34 5 39 35 5 39 49 3 39 50 3 39 51 3 39 54 3 39 55 3 39 56 3 39 57 3 40 32 5 40 48 3 40 51 3 40 52 3 40 53 3 40 54 3 41 26 5 41 27 5 41 29 5 41 30 5 41 31 5 41 47 3 41 49 3 41 52 3 41 55 3 41 56 3 41 57 3 41 58 3 42 28 5 42 29 5 42 30 5 42 32 5 42 33 5 42 34 5 42 50 3 42 53 3 42 54 3 42 55 3 42 56 3 43 26 5 43 27 5 43 30 5 43 31 5 43 49 3 43 52 3 43 53 3

43 54 3 43 55 3