package genetic;

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

int N=Integer.valueOf(args[0]);

int [] costsloev = new int[N];

int K=Integer.valueOf(args[1]);

int [] vertices =new int[K];

int [][] summatrixo=new int[K+1][K+1];

int[][] chromas = new int[N][N];

int minsum= Integer.MAX\_VALUE;

ArrayList<int[][]> matrsloev = zapolnenie(args,costsloev,vertices);

for (int i=0; i< summatrixo.length;i++){

for (int j=0; j< summatrixo[0].length;j++){

summatrixo[i][j]=0;

}

}

chromosome(chromas);

int MINsum = evaluationfunction(chromas,matrsloev,minsum,costsloev,K,vertices,summatrixo);

mutation(chromas);

int it=6;

for(int r=0; r<it;r++) {

MINsum = evaluationfunction(chromas, matrsloev, MINsum, costsloev, K, vertices, summatrixo);

mutation(chromas);

}

MINsum=evaluationfunction(chromas,matrsloev,MINsum,costsloev,K,vertices,summatrixo);

System.out.println("Матрица смежности для связного графа с дугами минимальной стоимости:");

for (int i=0; i< summatrixo.length;i++){

for (int j=0; j< summatrixo[0].length;j++){

System.out.print("\t"+summatrixo[i][j]);

}

System.out.println();

}

System.out.println("Сумма дуг:"+MINsum);

}

public static ArrayList<int[][]> zapolnenie(String [] args, int [] costsloev, int[] vertices ){

ArrayList<int[][]> matrsloev = new ArrayList<>();

int colvosl=Integer.valueOf(args[0]);

int ver=Integer.valueOf(args[1]);

int fig =0;

int gty=0;

for (int i=1; i< vertices.length+1;i++){

vertices[i-1]=i;

}

for (int i=0; i< costsloev .length;i++){

costsloev[i]=1+(int)(Math.random()\*100);

}

System.out.println("Матрица слоев ");

for(int b = 0; b < costsloev.length; b++){

System.out.print("\t"+costsloev[b]);

}

System.out.println();

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

boolean flag = true;

boolean flag1 = true;

int m = 4 + (int) (Math.random() \* (ver-4));

int[][] mas = new int[m + 1][m + 1];

for (int t = 0; t < mas.length; t++) {

for (int j = 0; j < mas[0].length; j++) {

mas[t][j] = 0;

}

}

for (int g = 1; g < m + 1; g++) {

mas[0][g] = 1 + (int) (Math.random() \* (ver));

do {

for (int p = 1; p < g + 1; p++) {

if ((mas[0][g] == mas[0][p]) & (g != p)) {

mas[0][g] = 1 + (int) (Math.random() \* (7 + 1));

p = 0;

}

}

flag = false;

} while (flag);

mas[g][0] = mas[0][g];

}

do{

for (int r = 1; r < mas.length; r++) {

for (int b = 1; b < mas.length; b++) {

if (r == b) {

mas[r][b] = 0;

} else {

if (r < b) {

mas[r][b] = (int) (Math.random() \* 2);

mas[b][r] = mas[r][b];

}

}

}

}

gty = 0;

for (int r = 1; r < mas.length; r++) {

for (int b = 1; b < mas.length; b++) {

if(mas[r][b]==0){

fig=fig +1;

}

}

if (fig==m){

gty=gty+1;

}

fig=0;

}

// System.out.println("число gty : "+gty);

if(gty==0){

flag1=false;

}

}while(flag1);

System.out.println("число m : "+m);

for(int r = 0; r < mas.length; r++){

for(int b = 0; b < mas.length; b++){

System.out.print("\t"+mas[r][b]);

}

System.out.println();

}

matrsloev.add(mas);

}

return matrsloev;

}

public static void chromosome(int [][] chromas){

for(int i=0; i< chromas.length; i++){

for(int j=0; j<chromas[0].length;j++){

chromas[i][j]=(int) ( Math.random() \* 2 );

}

}

System.out.println("Матрица популяции:");

for (int i=0; i< chromas.length;i++){

for (int j=0; j< chromas[0].length;j++){

System.out.print("\t"+chromas[i][j]);

}

System.out.println();

}

}

public static int evaluationfunction (int [][] chromas, ArrayList<int[][]> matrsloev, int minsum, int[] costsloev,int K,int[] vertices, int [][] summatrixo){

int[][] experimental=new int[K+1][K+1];

int [][] dostmstrix = new int[K+1][K+1];

int ind1=0;

int ind2=0;

int index1=0;

int index2=0;

for(int i=0; i< chromas.length; i++){

int sum = 0;

for (int l=0; l< experimental.length;l++){

for (int j=0; j< experimental[0].length;j++){

experimental[l][j]=0;

}

}

for (int l=1; l< experimental.length;l++){

experimental[0][l]=vertices[l-1];

}

for (int l=1; l< experimental.length;l++){

experimental[l][0]=vertices[l-1];

}

/\* System.out.println("Матрица смежности до заполнения:");

for (int l=0; l< experimental.length;l++){

for (int j=0; j< experimental[0].length;j++){

System.out.print("\t"+experimental[l][j]);

}

System.out.println();

}\*/

for (int j = 0; j < chromas[0].length; j++) {

if (chromas[i][j] == 1) {

int[][] gen = matrsloev.get(j);

for (int k = 1; k < gen.length; k++) {

for (int m = 1; m < gen.length; m++) {

if (gen[k][m] == 1) {

ind1=gen[k][0];

ind2=gen[0][m];

for (int y = 1; y < experimental.length; y++){

if(experimental[y][0]==ind1){

index1=y;

}

}

for (int y = 1; y < experimental.length; y++){

if(experimental[0][y]==ind2){

index2=y;

}

}

experimental[index1][index2]=1;

}

}

}

sum = sum + costsloev[j];

}

}

/\*System.out.println("Матрица смежности после заполнения:");

for (int l=0; l< experimental.length;l++){

for (int j=0; j< experimental[0].length;j++){

System.out.print("\t"+experimental[l][j]);

}

System.out.println();

}\*/

for (int l = 0; l < dostmstrix.length; l++){

dostmstrix[0][l]=0;

}

for (int l = 0; l < dostmstrix.length; l++){

dostmstrix[l][0]=0;

}

for (int l = 1; l < dostmstrix.length; l++) {

for (int g = 1; g < dostmstrix[0].length; g++) {

dostmstrix[l][g] = experimental[l][g];

}

}

for (int u=0;u< dostmstrix.length;u++){

for (int l = 1; l < dostmstrix.length; l++) {

for (int g = 1; g < dostmstrix[0].length; g++) {

if(dostmstrix[l][g]==1){

dostmstrix[0][g]=1;

dostmstrix[g][0]=1;

}

}

}

}

for(int k=1; k< dostmstrix.length; k++){

if(dostmstrix[k][0]==0){

sum=0;

}

}

int znach=i+1;

System.out.println("Вектор достежимости "+znach+" хромосомы:");

for (int l=1; l< dostmstrix.length;l++){

System.out.print("\t"+dostmstrix[l][0]);

}

System.out.println();

if(sum<minsum){

if(sum!=0) {

System.out.println("\t сумма "+sum);

System.out.println("\t минимальная сумма"+minsum);

minsum = sum;

for (int y = 0; y < experimental.length; y++) {

for (int u = 0; u < experimental[0].length; u++) {

summatrixo[y][u] = experimental[y][u];

}

}

System.out.println("Матрица смежности для минимальной суммы");

for (int y = 0; y < summatrixo.length; y++) {

for (int u = 0; u < summatrixo[0].length; u++) {

System.out.print("\t"+summatrixo[y][u]);

}

System.out.println();

}

}

}

}

return minsum;

}

public static void mutation(int [][] chromas){

int razm = chromas[0].length;

for(int i=0; i< chromas.length; i++){

int mutationPoint = (int)(Math.random()\*razm);

if (chromas[i][mutationPoint] == 0) {

chromas[i][mutationPoint] = 1;

} else {

chromas[i][mutationPoint] = 0;

}

mutationPoint = (int)(Math.random()\*razm);

if (chromas[i][mutationPoint] == 0) {

chromas[i][mutationPoint] = 1;

} else {

chromas[i][mutationPoint] = 0;

}

}

System.out.println("матрица после мутации:");

for (int i=0; i< chromas.length;i++){

for (int j=0; j< chromas[0].length;j++){

System.out.print("\t"+chromas[i][j]);

}

System.out.println();

}

}

/\*public static void crossover(int [][] chromas){

int[][] tmp = new int[chromas.length][chromas[0].length];

int Point = (int)(Math.random()\*chromas[0].length);

// System.out.println(" Граница "+Point);

for(int i=0; i< chromas.length; i++) {

int mam = (int)(Math.random()\*chromas.length);

// System.out.println(" первая особь"+mam);

int ded = (int)(Math.random()\*chromas.length);

//System.out.println(" вторая особь"+ded);

while (mam==ded){

ded=(int)(Math.random()\*chromas.length);

}

for(int j=0; j< chromas[0].length; j++){

if (j<Point){

tmp[i][j]=chromas[mam][j];

}

else tmp[i][j]=chromas[ded][j];

}

}

for(int i=0; i< chromas.length; i++) {

for(int j=0; j< chromas[0].length; j++){

chromas[i][j]=tmp[i][j];

}

}

System.out.println("матрица после селекции:");

for (int i=0; i< chromas.length;i++){

for (int j=0; j< chromas[0].length;j++){

System.out.print("\t"+chromas[i][j]);

}

System.out.println();

}

}\*/

}