/\*\*

數獨

\*/

package com.company;

import java.io.FileReader;

import java.io.\*;

import java.util.Scanner;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.lang.String;

public class Main {

public static void main(String[] args) {

String sudo\_str = read("sudo");

char[] sudo\_one\_arr = str\_to\_char(sudo\_str);

int[] all\_number\_sum = all\_number\_sum(sudo\_one\_arr);

int[] possible\_locate\_sum = possible\_locate\_sum(sudo\_one\_arr);

for(int i=0; i<9; i++){

System.out.println("數字" + (i+1) + "有：" + all\_number\_sum[i] + " 個");

System.out.println("數字" + (i+1) + "可填空格數：" + possible\_locate\_sum[i] + " 個");

}

for(int i=0; i<81; i++){

if(sudo\_one\_arr[i] == ' '){

int[] sum = find\_possible\_number\_in\_affect\_scope(sudo\_one\_arr,i);

System.out.print("第" + i + "格在影響範圍內可以填的數字：");

for(int j: sum){System.out.print(j + " ");}

System.out.print("\n");}

}

for(int j=0; j<81; j++){

if(sudo\_one\_arr[j] == ' '){

char[] list = judge(sudo\_one\_arr, j);

System.out.print("第" + j + "格可填：");

for(char i:list){System.out.print(i + " ");}

System.out.print("\n");}}

System.out.println(sudo\_one\_arr.length);

}

private static int char\_find(char[] char\_list, char search){

for(int i=0; i<char\_list.length; i++){

if(char\_list[i] == search){

return i;}}

return -1;

}

private static int char\_find\_sum(char[] char\_list, char search){

int sum = 0;

for(char i:char\_list){

if(i == search){sum++;}}

return sum;

}

private static int int\_find(int[] int\_list, int number){

int sum;

for(int i:int\_list){

if(i == number){

sum = i;

return sum;}}

return -1;

}

private static int[] find\_possible\_number\_in\_affect\_scope(char[] char\_list, int locate){

int x = (locate - locate%9);

int y = locate%9;

int x\_nine = ((x/9) - ((x/9)%3))/3;

int y\_nine = (y - y%3)/3;

int loc = x\_nine\*27 + y\_nine\*3;

int[] sum = {0,0,0,0,0,0,0,0,0};

char[] locate\_list = judge(char\_list,locate);

char[] number = {'1', '2', '3', '4', '5', '6', '7', '8', '9'};

int[] all\_locate = new int[81];

for(int i=0; i<81; i++){all\_locate[i] = 100;}

int j=0;

for(int i=0; i<9; i++){

if(int\_find(all\_locate,x+i) == -1){

all\_locate[j] = (x+i);

j++;}}

for(int i=0; i<9; i++){

if(int\_find(all\_locate,y + i\*9) == -1){

all\_locate[j] = (y + i\*9);

j++;}}

for(int i=0; i<3; i++){

if(int\_find(all\_locate,loc + 9\*i) == -1){

all\_locate[j] = (y + i\*9);

j++;}

if(int\_find(all\_locate,loc + 1 + 9\*i) == -1){

all\_locate[j] = (y + 1 + i\*9);

j++;}

if(int\_find(all\_locate,loc + 2 + 9\*i) == -1){

all\_locate[j] = (y + 2 + i\*9);

j++;}}

for(int i:all\_locate){

if(i != 100){

char[] possible\_list = judge(char\_list,i);

for(char k:possible\_list){

if(char\_find(locate\_list,k) != -1){

loc = char\_find(number,k);

sum[loc]++;}}}}

return sum;

}

private static int[] possible\_locate\_sum(char[] char\_list){

int[] sum = {0,0,0,0,0,0,0,0,0};

char[] number = {'1', '2', '3', '4', '5', '6', '7', '8', '9'};

for(int i=0; i<9; i++){

for(int j=0; j<81; j++) {

if(char\_list[j] == ' ' && char\_find(judge(char\_list,j),number[i]) != -1){

sum[i]++;}}}

return sum;

}

private static void print\_the\_sudo(char[] char\_list){

for(int i=0; i<81; i++){

System.out.print(char\_list[i]);

if(i%9 == 8){System.out.print("\n");}

}

}

private static char[] str\_to\_char(String sudo\_str){

char[] sudo\_one\_arr\_temp = sudo\_str.toCharArray();

char[] sudo\_one\_arr = new char[81];

for(int i=0,j=0; i<sudo\_one\_arr\_temp.length; i++){

if(sudo\_one\_arr\_temp[i] != '\n'){

sudo\_one\_arr[j] = sudo\_one\_arr\_temp[i];

j++;}}

return sudo\_one\_arr;

}

private static int[] all\_number\_sum(char[] char\_list){

int[] number\_sum = new int[9];

char[] number\_arr = {'1', '2', '3', '4', '5', '6', '7', '8', '9'};

for(int i=0; i<9; i++){

number\_sum[i] = char\_find\_sum(char\_list, number\_arr[i]); }

return number\_sum;

}

private static String read(String file\_name){

StringBuilder content = new StringBuilder();

try{

FileReader file\_reader = new FileReader(file\_name + ".txt");

BufferedReader buffer\_reader = new BufferedReader(file\_reader);

String fr;

while((fr= buffer\_reader.readLine()) != null){

content.append(fr + '\n');

}

}

catch(IOException error){System.out.println(error);}

String contentI = content.toString();

return contentI;

}

private static void write(String file\_name, String content){

try{

FileWriter file\_writer = new FileWriter(file\_name + ".txt");

BufferedWriter buffer\_writer = new BufferedWriter(file\_writer);

buffer\_writer.write(content);

buffer\_writer.flush();

buffer\_writer.close();

}

catch(IOException error){System.out.println(error);}

}

private static char[] judge(char[] one\_arr, int locate){

StringBuilder horizontal\_gap = new StringBuilder();

int x = (locate - locate%9);

char[] horizontal = new char[9];

for(int i = x; i<(x+9); i++){

horizontal[i%9] = one\_arr[i];}

char[] number = {'1', '2', '3', '4', '5', '6', '7', '8', '9'};

for(char i:number){

if(char\_find(horizontal,i) == -1){

horizontal\_gap.append(i);}}

char[] horizontal\_gap\_arr = horizontal\_gap.toString().toCharArray();

StringBuilder vertical\_gap = new StringBuilder();

int y = locate%9;

char[] vertical = new char[9];

for(int i=0; i<9; i++ ){

vertical[i] = one\_arr[y + (i\*9)];}

for(char i:number){

if(char\_find(vertical,i) == -1){

vertical\_gap.append(i);}}

char[] vertical\_gap\_arr = vertical\_gap.toString().toCharArray();

int x\_nine, y\_nine;

x\_nine = ((x/9) - ((x/9)%3))/3;

y\_nine = (y - y%3)/3;

int loc = x\_nine\*27 + y\_nine\*3;

char[] nine\_nine = new char[9];

for(int i=0; i<3; i++){

nine\_nine[i\*3] = one\_arr[loc + 9\*i];

nine\_nine[i\*3 + 1] = one\_arr[loc + 1 + 9\*i];

nine\_nine[i\*3 + 2] = one\_arr[loc + 2 + 9\*i];

}

StringBuilder nine\_nine\_gap = new StringBuilder();

for(char i:number){

if(char\_find(nine\_nine, i) == -1){nine\_nine\_gap.append(i);}}

char[] nine\_nine\_gap\_arr = nine\_nine\_gap.toString().toCharArray();

StringBuilder total\_gap = new StringBuilder();

for(char i:number){

if(char\_find(horizontal\_gap\_arr, i) != -1 && char\_find(vertical\_gap\_arr, i) != -1 && char\_find(nine\_nine\_gap\_arr, i) != -1){

total\_gap.append(i);}}

char[] total\_gap\_arr = total\_gap.toString().toCharArray();

return total\_gap\_arr;

}

}