import java.io.\*;

import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.List;

import java.util.Map.Entry;

import java.util.Set;

public class Hua {

static HashMap<String, Number> map = new HashMap<>();

public static void initSys() {

map.clear();

}

public static String loadNumbers(String fileName) {

if(fileName == null || fileName.isEmpty()) return "E002:入参输入异常";

if(!fileName.endsWith(".txt")) {

return "E004:文件格式不正确";

}

try {

File file = new File(fileName);

if (file.isFile() && file.exists()) {

InputStreamReader isr = new InputStreamReader(new FileInputStream(file), "utf-8");

BufferedReader br = new BufferedReader(isr);

String lineTxt = null;

while ((lineTxt = br.readLine()) != null) {

String[] s = lineTxt.split("\\|");

Number n = new Number(s[0].trim(), s[1].trim(), s[2].trim(), s[3].trim());

map.put(s[0], n);

}

br.close();

return "E001:成功";

} else {

return "E003:文件不存在";

}

} catch (Exception e) {

return "E000:方法未实现";

}

}

public static String queryNumber(String id, float price, String brand, int status) {

StringBuilder sb = new StringBuilder();

List<Long> ls = new ArrayList<>();

Set<Entry<String, Number>> set = map.entrySet();

for(Entry<String, Number> entry : set ) {

boolean flag = true;

Number n = entry.getValue();

if(!id.isEmpty()) flag = n.getId().equals(id);

**//浮点型相等进行判断**

if(**price - 0 >=1e-7**) flag = Float.parseFloat(n.getPrice()) - price <= 1e-7;

if(!brand.equals("-1")) flag = n.getBrand().equals(brand) || n.getBrand().equals("0");

flag = status == Integer.parseInt(n.getStatus());

if(flag) ls.add(Long.parseLong(n.getId()));

}

if(ls.size() == 0) return "";

else {

Collections.sort(ls);

for(int i = 0; i<ls.size()-1; i++) {

sb.append(ls.get(i)+";");

}

sb.append(ls.get(ls.size()-1));

return sb.toString();

}

}

public static String randomPick(String brand, int randNumber) {

StringBuilder sb = new StringBuilder();

List<Long> ls = new ArrayList<>();

Set<Entry<String, Number>> set = map.entrySet();

for(Entry<String, Number> entry : set ) {

Number n = entry.getValue();

boolean flag = n.getStatus().equals("0");

if(!brand.equals("-1")) flag = n.getBrand().equals(brand) || n.getBrand().equals("0");

if(flag) ls.add(Long.parseLong(n.getId()));

}

if(ls.size() == 0) return "";

else if(ls.size() <= randNumber) {

Collections.sort(ls);

for(int i = 0; i<ls.size()-1; i++) {

sb.append(ls.get(i)+";");

}

sb.append(ls.get(ls.size()-1));

return sb.toString();

}else {

//随机扔掉多余的

int k = ls.size() - randNumber ;

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

ls.remove((int)Math.random()\*ls.size());

}

Collections.sort(ls);

for(int i = 0; i<ls.size()-1; i++) {

sb.append(ls.get(i)+";");

}

sb.append(ls.get(ls.size()-1));

return sb.toString();

}

}

public static String selfServicePick(String brand, String expression1, String expression2) {

StringBuilder sb = new StringBuilder();

List<Long> ls = new ArrayList<>();

Set<Entry<String, Number>> set = map.entrySet();

for(Entry<String, Number> entry : set ) {

boolean flag = true;

Number n = entry.getValue();

if(!brand.equals("0")) flag = n.getBrand().equals(brand);

String id = n.getId();

if(!expression1.isEmpty() && flag) {

for(int i = 0; i < id.length() && flag;i++) {

if(expression1.charAt(i) == '\*') continue;

else {

if(id.charAt(i) != expression1.charAt(i)) {

flag = false;

break;

}

}

}

}

if(!expression2.isEmpty() && flag) {

List<Character> list = new ArrayList<>();

for(int i = expression2.length()-1;i >= 0; i-=2) {

list.add(expression2.charAt(i));

}

for(int i = 0; i < list.size(); i++){

if(list.get(i) == id.charAt(id.length()-1-i)) {

flag = false;

break;

}

}

}

if(flag) ls.add(Long.parseLong(n.getId()));

}

if(ls.size() == 0) return "";

else {

Collections.sort(ls);

for(int i = 0; i<ls.size()-1; i++) {

sb.append(ls.get(i)+";");

}

sb.append(ls.get(ls.size()-1));

return sb.toString();

}

}

public static String lockNumber(String id, String customer) {

try {

Number n = map.get(id);

if(n == null) return "E007:锁定号码在系统中不存在";

if(n.getStatus().equals("1")) return "E005:号码不能重复锁定";

int count = 0;

Set<Entry<String, Number>> set = map.entrySet();

for(Entry<String, Number> entry : set ) {

Number temp = entry.getValue();

if(temp.getStatus().equals("1") && temp.getCustomer() != null && temp.getCustomer().equals(customer)) count++;

}

if(count >= 2) return "E006:客户锁定超过两个号码";

n.setCustomer(customer);

n.setStatus("1");

n.setTime(System.currentTimeMillis()/1000 + 24\*60\*60);

return "E000:成功";

}catch(Exception e) {

return "E000:方法未实现";

}

}

public static String unlockNumber() {

try {

Set<Entry<String, Number>> set = map.entrySet();

for(Entry<String, Number> entry : set ) {

Number n = entry.getValue();

if(n.getStatus().equals("1") && n.getTime() - System.currentTimeMillis()/1000 >24\*60\*60L) {

n.setStatus("0");

n.setTime(0L);

}else {

continue;

}

}

return "E001:成功";

}catch(Exception e) {

return "E000:方法未实现";

}

}

public static void main(String[] args) {

String filePath = "D:\\Java\\huawei\\numbers.txt";

loadNumbers(filePath);

//System.out.println(queryNumber("", 0f, "-1", 0));

// System.out.println(randomPick("-1", 2));

//System.out.println(selfServicePick("0", "152\*\*\*\*\*\*\*\*", "!0"));

System.out.println(lockNumber("15261812225", "2"));

}

}