第3次实验：

1. 编程在一个一维整数数组中查找指定的整数出现了几次，各在何下标位置。

**import** java.util.\*;

**public** **class** Hello{

**public** **static** **void** main(String[] args){

**int** bound = 10;

**int**[] a=**new** **int**[50];

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

a[i] = (**int**)(Math.*random*()\*bound);

}

System.*out*.println("array:");

System.*out*.println(Arrays.*toString*(a));

Scanner sc =**new** Scanner(System.*in*);

System.*out*.printf("请输入在数组中所要查找的数, 范围(0~"+(bound-1)+"):");

**int** x=sc.nextInt();

**int** cnt=0;

**for**(**int** j=0;j<a.length;j++){

**if**(a[j]==x){

System.*out*.printf("%d ",j);

cnt++;

}

}

**if**(cnt==0) System.*out*.println("没有"+x);

**else** System.*out*.printf("有%d个%d。",cnt,x);

}

}

运行结果：

array:

[6, 4, 5, 9, 3, 9, 5, 8, 0, 6, 0, 0, 9, 5, 0, 8, 7, 1, 8, 0, 5, 2, 4, 5, 6, 3, 1, 2, 7, 1, 9, 1, 9, 8, 8, 0, 3, 3, 8, 1, 9, 2, 0, 2, 7, 7, 1, 2, 5, 8]

请输入在数组中所要查找的数, 范围(0~9):0

8 10 11 14 19 35 42 有7个0。

1. 判断一个二维数组是否是对角矩阵、上三角矩阵、下三角矩阵和单位矩阵。

**import** java.util.\*;

**public** **class** mat {

**public** **static** **void** main(String[] args) {

**int** a[][][] = { { { 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 } },

{ { 0, 1, 2 }, { 0, 4, 5 }, { 0, 0, 8 } },

{ { 0, 0, 0 }, { 3, 4, 0 }, { 6, 7, 8 } },

{ { 1, 0, 0 }, { 0, 1, 0 }, { 0, 0, 1 } },

{ { 1, 0, 0 }, { 0, 2, 0 }, { 0, 0, 1 } }};

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

System.*out*.println("test case " + i);

**for** (**int** j = 0; j < a[i].length; j++) {

System.*out*.println(Arrays.*toString*(a[i][j]));

}

**int**[][] arr = a[i];

**boolean** duijiao = **true**, shangsanjiao = **true**, xiasanjiao = **true**, danwei = **true**;

**for** (**int** k = 0; k < arr.length; k++)

**for** (**int** j = 0; j < arr[k].length; j++) {

**if** (k != j) {

**if** (arr[k][j] != 0) {

duijiao = **false**;

danwei = **false**;

}

}

**if** (danwei && k == j) {

**if** (arr[k][j] != 1)

danwei = **false**;

}

**if** (shangsanjiao && k > j) {

**if** (arr[k][j] != 0)

shangsanjiao = **false**;

}

**if** (xiasanjiao && k < j) {

**if** (arr[k][j] != 0)

xiasanjiao = **false**;

}

}

**if** (danwei)

System.*out*.println("单位阵");

**else** **if** (duijiao)

System.*out*.println("对角阵");

**else** **if** (shangsanjiao)

System.*out*.println("上三角阵");

**else** **if** (xiasanjiao)

System.*out*.println("下三角阵");

**else**

System.*out*.println("普通矩阵");

System.*out*.println();

}

}

}

运行结果：

test case 0

[0, 1, 2]

[3, 4, 5]

[6, 7, 8]

普通矩阵

test case 1

[0, 1, 2]

[0, 4, 5]

[0, 0, 8]

上三角阵

test case 2

[0, 0, 0]

[3, 4, 0]

[6, 7, 8]

下三角阵

test case 3

[1, 0, 0]

[0, 1, 0]

[0, 0, 1]

单位阵

test case 4

[1, 0, 0]

[0, 2, 0]

[0, 0, 1]

对角阵

1. 编程将一个字符串一分为二，一个存放其中的汉字，另一个是其他字符。

**import** java.util.\*;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** hanzi {

**public** **static** **void** main(String[] args) {

Pattern p\_str = Pattern.*compile*("[\\u4e00-\\u9fa5]+");

String str = "字符串：やめてmi汉cmiu.co哈哈^&\*()\_+{";

System.*out*.println(str);

String hanzi = "";

String zifu = "";

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

**char** ch = str.charAt(i);

String s = ch + "";

Matcher m = p\_str.matcher(s);

**if**(m.find()&&m.group(0).equals(s)){

hanzi += ch;

}**else**

zifu += ch;

}

System.*out*.println("汉字：");

System.*out*.println(hanzi);

System.*out*.println("除了汉字外的字符：");

System.*out*.println(zifu);

}

}

运行结果：

字符串：やめてmi汉cmiu.co哈哈^&\*()\_+{

汉字：

字符串汉哈哈

除了汉字外的字符：

：やめてmicmiu.co^&\*()\_+{

4. 使用Vector向量类完成一个整数集合的增、删、改、查功能。

比如：程序运行起来后，

(1)输入1表示增加,再输入234,则增加234到集合中

(2)输入2表示删除,再输入3,则删除集合中的第3个数

(3)输入3表示修改,再输入2,567,则将集合中的第2个数改为567

(4)输入4表示查找,再输入789,则在集合中查找是否存在

**import** java.util.\*;

**public** **class** set {

**public** **static** **void** main(String[] args) {

Vector<Integer> v = **new** Vector<Integer>(10,2);

**for** (**int** i=0;i<4;i++){

v.add((**int**)(Math.*random*()\*10));

}

System.*out*.println("整数集合初始化为：");

*printset*(v);

System.*out*.println();

**int** operation=0;

**while**(operation!=-1){

System.*out*.println();

System.*out*.println("1:增加 2:删除 3: 修改 4:查找");

System.*out*.println();

System.*out*.printf("操作：");

Scanner sc =**new** Scanner(System.*in*);

operation=sc.nextInt();

**if**(operation==1){

System.*out*.printf("增加的元素：");

v.add(sc.nextInt());

}

**else** **if**(operation==2){

System.*out*.printf("删除第几个元素：");

v.remove(sc.nextInt());

}

**else** **if**(operation==3){

System.*out*.printf("修改第几个元素, 为几, 以空格分开");

**int** a = sc.nextInt();

**int** b = sc.nextInt();

v.setElementAt(b,a);

}

**else** **if**(operation==4){

System.*out*.printf("查找的元素：");

**int** i =v.indexOf(sc.nextInt());

**if**(i!=-1)

System.*out*.println("下标为"+i);

**else**

System.*out*.println("没有你查找的元素");

}

**else** {

operation=-1;

System.*out*.println("error");

**return**;

}

System.*out*.println("现在的整数集合为：");

*printset*(v);

}

}

**public** **static** **void** printset(Vector<Integer> v){

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

System.*out*.print(i+": "+v.get(i)+", ");

}

}

}

运行结果：

整数集合初始化为：

0: 6, 1: 2, 2: 2, 3: 3,

1:增加 2:删除 3: 修改 4:查找

操作：1

增加的元素：234

现在的整数集合为：

0: 6, 1: 2, 2: 2, 3: 3, 4: 234,

1:增加 2:删除 3: 修改 4:查找

操作：2

删除第几个元素：3

现在的整数集合为：

0: 6, 1: 2, 2: 2, 3: 234,

1:增加 2:删除 3: 修改 4:查找

操作：3

修改第几个元素, 为几, 以空格分开2 567

现在的整数集合为：

0: 6, 1: 2, 2: 567, 3: 234,

1:增加 2:删除 3: 修改 4:查找

操作：4

查找的元素：789

没有你查找的元素

现在的整数集合为：

0: 6, 1: 2, 2: 567, 3: 234,

1:增加 2:删除 3: 修改 4:查找

操作：4

查找的元素：234

下标为3

现在的整数集合为：

0: 6, 1: 2, 2: 567, 3: 234,

1:增加 2:删除 3: 修改 4:查找