浦发机试题：

1. 三个数排序

**public** **static** **void** sort(**int** a,**int** b,**int** c) {

**int** temp = 0;

**if**(a>b) {

temp = a;

a = b;

b = temp;

}

**if**(a>c) {

temp = a;

a = c;

c = temp;

}

**if**(b>c) {

temp = b;

b = c;

c = temp;

}

}

1. 字符串倒序

**public** **static** String revedsWords(String s) {

String [] parts = s.trim().split("\\s+");

String out = "";

**for**(**int** i = parts.length-1;i>0;i--) {

out += parts[i]+" ";

}

**return** out+parts[0];

}

1. 最大序列求和

**public** **static** **int** FindGreatestSumOfSubArray(**int** [] array) {

**int** res = array[0];

**int** max = array[0];

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

max = Math.*max*(max+array[i], array[i]);

res = Math.*max*(max, res);

}

**return** res;

}

1. 字符串反转

**public** **static** String reverdsWords(String s) {

String [] parts = s.split(" ");

StringBuilder sb = **new** StringBuilder();

**for**(String a:parts) {

StringBuilder temp = **new** StringBuilder(a);

sb.append(temp.reverse());

sb.append(" ");

}

**return** sb.toString().trim();

}

1. 数列求和

sum+=sum;

1. 字符串缩写

1. 分段函数实现
2. 两个字符串查重

1. 凯撒密码

**public** **static** **void** Decrypt(String str,**int** n) {

//解密,加密同理

**int** k = Integer.*parseInt*("-"+n);

String string="";

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

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

**if**(c>='a'&&c<='z') {

c += k%26;

**if**(c<'a') {

c+=26;

}

**if**(c>'z') {

c-=26;

}

}

**else** **if**(c>='A'&&c<='Z') {

c+=k%26;

**if**(c<'A') {

c+=26;

}

**if**(c>'Z') {

c-=26;

}

}

string = string +c;

}

System.***out***.println(str+"解密后为"+string);

}

1. 从数组中找出唯一出现一次的数

**public** **static** **void** FindNumsAppearOnce(**int** [] array,**int** num1[],**int** num2[]) {

Map<Integer,Integer> map = **new** HashMap<Integer,Integer>();

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

**if**(map.containsKey(array[i])) {

**int** time = map.get(array[i]);

map.put(array[i], ++time);

}**else** map.put(array[i], 1);

}

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

**if**(map.get(array[j])==1&&num1[0]==0) {

num1[0]=array[j];

}

**else** **if**(map.get(array[j])==1&&num1[0]!=0) {

num2[0]=array[j];

}

}

}

1. 给年月日，判断是这一年的第几天，这一周的第几天

**public** **static** **void** orderDate(String date) {

**int** dateSum = 0;

**int** year = Integer.*valueOf*(date.substring(0,4));

**int** month = Integer.*valueOf*(date.substring(5,7));

**int** day = Integer.*valueOf*(date.substring(8,10));

**for**(**int** i=1;i<month;i++) {

**switch**(i) {

**case** 1:

**case** 3:

**case** 5:

**case** 7:

**case** 8:

**case** 10:

**case** 12:

dateSum+=31;

**break**;

**case** 4:

**case** 6:

**case** 9:

**case** 11:

dateSum+=30;

**break**;

**case** 2:

**if**((year%4==0&&year%100!=0)||(year%400==0)){

dateSum+=29;

}**else** {

dateSum+=28;

}

}

}

System.***out***.println("您输入的日期是该年的第："+(dateSum+day)+"天");

}

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

**while**(**true**){

System.***out***.println("请输入年月日(格式2015-02-11)：");

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

String str=sc.nextLine();

Order.*orderDate*(str);

}

}

1. 小球从100米下落，每次回弹一半距离，第几次落地后的总距离

**public** **static** **void** high(**int** n) {

**double** j = 50;

**double** sum = 0;

**for**(**int** i =1;i<=n;i++) {

**if**(i==1) {

sum = 100;

}

**else** {

sum += j\*2;

j = j/2;

}

}

System.***out***.println(sum);

System.***out***.println(j);

}

1. 三维数组求对角线上的元素之和
2. 对于若干字符和数字的字符串，求其中所有数字之和。

**public** **static** **int** numSUm(String str) {

**char**[] c = str.toCharArray();

**int** sum = 0;

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

**if**(c[i]>='0'&&c[i]<='9') {

sum += (c[i]-'0');

}

}

System.***out***.println(sum);

**return** sum;

}

1. 利用快速排序实现所有奇数在前，偶数在后

**public** **static** **void** fun(**int** [] arr) {

**int** front = 0;

**int** end = arr.length-1;

**if**(arr.length==0) {

**return** ;

}

**while** (front < end) {

**while**(front<arr.length&&arr[front]%2 ==1) {

front ++;

}

**while**(end>=0 &&arr[end]%2==0) {

end --;

}

**if**(front < end ) {

**int** temp = arr[front];

arr[front]= arr[end];

arr[end]=temp;

}

}

}

1. 数组元素交换位置
2. 约瑟夫环

**public** **static** **int** getResult(**int** n ,**int** m) {

//n代表游戏人数，m代表数到m出局，求最后一个出局人的编号

LinkedList<Integer> test = **new** LinkedList<Integer>();

**for**(**int** i=1;i<=n;i++) {

test.add(i);

}

**int** start = 0;

**while**(test.size()>1) {

**int** need = (start+m-1)%test.size();

test.remove(need);

start = need%test.size();

}

**return** test.get(0);

}

1. 1+2/3+3/5+4/7+….

**public** **static** **double** fenshusum(**int** n) {

**double** sum = 0;

**for**(**int** i=1;i<=n;i++) {

sum += i/(2\*i-1);

}

**return** sum;

}

1. 给出一个字符串，由不同的单词用空格隔开，然后把这些单词的首字母取出并大写输出

**public** **static** String Daxie(String s) {

**char** [] c = s.toCharArray();

StringBuilder sb = **new** StringBuilder();

sb.append(c[0]);

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

**if**(c[i]==' ') {

sb.append(c[i+1]);

}

}

String str = sb.toString();

String strs = str.toUpperCase();

System.***out***.println(strs);

**return** strs;

}

1. 猴子吃桃

**public** **static** **int** taozi(**int** n) {

**int** m =1 ;

**for**(**int** i = 1;i<n;i++) {

m = (m+1)\*2;

}

System.***out***.println(m);

**return** m;

}

1. 输入一段话，输出字的个数
2. 非完全平方数的判断
3. 找出正整数中的偶数，并输出相加后的数，用long，5548，12

**public** **static** **int** oushu(**long** n) {

String s = String.*valueOf*(n);

**char** [] c = s.toCharArray();

**int** sum = 0;

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

**if**(c[i]%2==0) {

sum = sum + (c[i]-'0');

}

}

System.***out***.println(sum);

**return** sum;

}

1. 输入n和b，找出1到n中被b整除的个数，输入6，3输出2
2. 爬一个或2个台阶，输入1<=n<90的数，以输入0作为结束标志

浦发java题

1. 数据库，写查询语句
2. Java内存泄露如何解决
3. Linux基础命令
4. Java clone
5. 判断带头节点的单向链表是否有环
6. 求一个数的反码
7. 数据库主键，连接方式有哪些
8. 求字符串的子串个数
9. 数据库事务，隔离级别
10. 数组和链表的区别
11. 数据库的触发器