3.分析以下需求，并用代码实现：

(1)倒着打印九九乘法表

public class Demo01 {

public static void main(String[] args) {

for(int i=9;i>0;i--){

for(int j=1;j<=i;j++){

System.out.print("i"+"\*"+"j"+"="+i\*j+" ");

}

System.out.println();

}

}

}

5.分析以下需求，并用代码实现：

(1)定义一个int类型的一维数组，内容为{6,2,9,15,1,5,20,7,18}

(2)将数组最大元素与最后一位元素进行交换,最小元素与第一位元素进行交换，并打印数组

提示思路：先查找最大值和最小值出现的索引。

public class Demo {

public static void main(String[] args) {

int [] arr ={6,2,9,15,1,5,20,7,18};

int max = arr[0];

int min = arr[0];

int d =0;

int x = 0;

for (int i = 0; i < arr.length ;i ++ ) { //遍历出最大值和最小值

if (max < arr[i]) {

max = arr[i];

}else if (min > arr[i]) {

min = arr[i];

}

}

//根据最大值,和最小值进行搜索,找出相对应的索引值

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

if (max == arr[i]) {

d = i;

}else if (min == arr[i]) {

x = i;

}

}

//打印出最大值最小值以及相对应的索引

System.out.println(max);

System.out.println(min);

System.out.println(d);

System.out.println(x);

//取得了最大值和最小值的索引,进行交换;

int temp = arr[d];

arr[d] = arr[arr.length -1];

arr[arr.length-1] = temp;

temp = arr[x];

arr[x] = arr[0];

arr[0] = temp;

//打印新的数组

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

System.out.print(arr[i] + " ");

}

}

}

6.分析以下需求，并用代码实现：（扩展）

(1)键盘录入6个int类型的数据存数数组arr中

(2)将arr数组中的内容反转

(3)将翻转后的数组角标为奇数的互相交换 1和3换, 3和5换,以此类推

(4)最后将数组最后一个角标为奇数的元素 和数组中第一个角标为奇数的元素交换

(5)打印最终的数组(实现了1-4步之后的数组)

(6)如：用户输入的6个int数字为[1,2,3,4,5,6],最后输出的结果为[6, 5, 4, 1, 2, 3]

import java.util.Scanner;

public class HomeWork01 {

/\*(1)键盘录入6个int类型的数据存数组arr中

(2)将arr数组中的内容反转

(3)最后将数组最后一个角标为奇数的元素 和数组中第一个角标为奇数的元素交换

(4)将第三步得到的数组进行排序。

(5)打印最终的数组(实现了1-4步之后的数组)\*/

public static void main(String[] args){

//定义数组

int[] arr=new int[6];

//调用向数组中添加元素方法

addElements(arr);

System.out.print("向数组中添加元素后输出数组中元素:");

printArray(arr);

//调用逆序数组元素方法

reverseArray(arr);

System.out.print("逆序数组中元素后输出数组中元素:");

printArray(arr);

//调用交换数组中元素方法

swapArrayElement(arr);

System.out.print("交换数组中元素后输出数组中元素:");

printArray(arr);

//调用选择排序方法进行排序

selectSort(arr);

System.out.print("进行选择排序后输出数组中元素:");

printArray(arr);

//调用选择排序方法进行排序

bubbleSort(arr);

System.out.print("进行冒泡排序后输出数组中元素:");

printArray(arr);

}

//定义键盘输入六个整数存储在数组中

public static void addElements(int[] arr){

Scanner sc=new Scanner(System.in);

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

System.out.print("请输入第"+(i+1)+"个整数:");

arr[i]=sc.nextInt();

}

}

//定义逆转数组元素方法

public static void reverseArray(int[] arr){

int temp=0;

for(int i=0,j=arr.length-1;i<arr.length/2;i++,j--){

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

//交换数组中元素 最后将数组最后一个角标为奇数的元素 和数组中第一个角标为奇数的元素交换

//第一个角标为奇数的元素是arr[1]; 偶数个元素则最后一个元素的角标是奇数arr[arr.lenrth-1], 奇数个元素则最后一个元素的角标是 arr[arr.length-2]

public static void swapArrayElement(int[] arr){

if(arr.length%2==0){ //偶数个元素

int temp=arr[1];

arr[1]=arr[arr.length-1];

arr[arr.length-1]=temp;

}

else{ //奇数个元素

int temp=arr[1];

arr[1]=arr[arr.length-2];

arr[arr.length-2]=temp;

}

}

//定义打印输出数组元素的方法 格式化输出

public static void printArray(int[] arr){

System.out.print("[");

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

if(i<arr.length-1){

System.out.print(arr[i]+",");

}else{

System.out.println(arr[i]+"]");

}

}

}

//选择排序

public static void selectSort(int[] arr){

for(int i=0;i<arr.length-1;i++){

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

if(arr[i]>arr[j]){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

}

//冒泡排序法

public static void bubbleSort(int[] arr){

for(int i=0;i<arr.length-1;i++){

for(int j=0;j<arr.length-i-1;j++){

if(arr[j]>arr[j+1]){

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

}

2.定义一个int型的一维数组，包含10个元素，分别赋一些随机整数，

然后求出所有元素的最大值，最小值，平均值，和值，并输出出来。

import java.util.Random;

public class HomeWork02 {

public static void main(String[] args){

//定义数组

int[] arr=new int[6];

//调用向数组中添加元素方法

addElements(arr);

System.out.println("向数组中添加元素后输出数组中元素:");

printArray(arr);

//调用求数组元素中最大值方法

int max=getMax(arr);

System.out.println("数组元素中最大值:"+max);

//调用求数组元素中最小值方法

int min=getMin(arr);

System.out.println("数组元素中最小值:"+min);

//调用数组求平均值方法

double dAvg=getAvg(arr);

System.out.println("数组中所有元素的平均值:"+dAvg);

//调用数组求和方法

int iSum=getSum(arr);

System.out.println("数组中所有元素的和:"+iSum);

}

//定义随机生成10整数存储在数组中

public static void addElements(int[] arr){

Random ran=new Random();

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

arr[i]=ran.nextInt(100);

}

}

//定义求数组元素最大值方法

public static int getMax(int[] arr){

selectSort(arr);

return arr[arr.length-1];

}

//定义求数组元素最小值方法

public static int getMin(int[] arr){

selectSort(arr);

return arr[0];

}

//定义求数组中元素平均值方法

public static double getAvg(int[] arr){

int sum=0;

for(int i=0;i<arr.length-1;i++){

sum+=arr[i];

}

return sum/arr.length;

}

//定义求数组中所有元素和方法

public static int getSum(int[] arr){

int sum=0;

for(int i=0;i<arr.length-1;i++){

sum+=arr[i];

}

return sum;

}

//排序数组

//选择排序法

public static void selectSort(int[] arr){

for(int i=0;i<arr.length-1;i++){

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

if(arr[i]>arr[j]){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

}

//冒泡排序法

public static void bubbleSort(int[] arr){

for(int i=0;i<arr.length-1;i++){

for(int j=0;j<arr.length-i-1;j++){

if(arr[j]>arr[j+1]){

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

//定义打印输出数组元素的方法 格式化输出

public static void printArray(int[] arr){

System.out.print("[");

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

if(i<arr.length-1){

System.out.print(arr[i]+",");

}else{

System.out.println(arr[i]+"]");

}

}

}

}

3.求1-500之内能同时被2，5，7整除的所有数。

import java.util.Random;

public class Demo01{

public static void main(String[] args) {

for(int i=0;i<=500;i++){

if(i%2==0&&i%5==0&&i%7==0){

System.out.println(i);

}

}

}

}

求100以内能被3整除的数加上能被5整除的数的和。

Public class Demo{

public static void main(String[] args){

getSum();

}

public static void getSum(){

int sum1=0;

int sum2=0;

int sum=0;

for(int i=1;i<=100;i++){

if(i%3==0){ //能同被3整除的数的和

sum1+=i;

}

if(i%5==0){ //能同被5整除的数的和

sum1+=i;

}

}

sum=sum1+sum2;

System.out.println("100以内能被3整除的数加上能被5整除的数的和是:"+sum);

}

}