Sorting

class Sorting{

//BUBBLE SORTING  
 int[] bubble(int[] arr){  
 int n=arr.length;  
 for(int i=0;i<n-1;i++){//-n  
 for(int j=i+1;j<n;j++){//--n\*n  
 if(arr[i]>arr[j]){  
 swap(arr,i,j);  
 }  
 }  
 }  
  
 return arr;  
 }

//SELECTION SORTING  
 int[] selection(int[] arr){  
 int n=arr.length;  
 for(int i=0;i<n;i++){  
 int last=n-i-1;  
 int maxindex=MAXOF(arr,0,last);  
 swap(arr,maxindex,last);  
 }  
 return arr;  
 }  
//INSERTION SORTING  
 int[] insertion(int[] arr){  
 int n=arr.length;  
 for(int i=0;i<n-1;i++){  
 int j=i+1;  
 while(j>0){  
 if(arr[j]>=arr[j-1]){  
 break;  
 }  
 else{  
 swap(arr,j,j-1);  
 j--;  
 }  
 }  
 }  
 return arr;  
 }  
//CYCLIC SORTING  
 int[] Cyclic(int[] arr){  
 int i=0;  
 while(i<arr.length){  
 if(i==arr[i]-1){  
 i++;  
 }  
 else{  
 swap(arr,i,arr[i]-1);  
 }  
 }  
  
 return arr;  
 }  
  
 private void swap(int[] arr, int first, int second) {  
 int temp=arr[first];  
 arr[first]=arr[second];  
 arr[second]=temp;  
 }  
  
 private int MAXOF(int[] arr, int start, int end) {  
 int max=start;  
 for(int i=0;i<=end;i++){  
 if(arr[i]>arr[max]  
 ){  
 max=i;  
 }  
 }  
 return max;  
  
 }  
  
}

//MERGER-INPLACE SORTING

public void mergesort(int[] arr, int s,int e){  
 if(s>=e){  
 return;  
 }  
 int mid=s+(e-s)/2;  
 mergesort(arr,s,mid);  
 mergesort(arr,mid+1,e);  
 mereger(arr,s,mid,e);  
}  
  
private void mereger(int[] arr, int s, int mid, int e) {  
 int idx=s;  
 int idx2=mid+1;  
 int k=0;  
 int[] mix=new int[e-s+1];  
 while(idx<=mid && idx2<=e){  
 if(arr[idx]<=arr[idx2]){  
 mix[k++]=arr[idx++];  
  
 }  
 else{  
 mix[k++]=arr[idx2++];  
 }  
 }  
 while(idx<=mid){  
 mix[k++]=arr[idx++];  
 }  
 while(idx2<=e){  
 mix[k++]=arr[idx2++];  
 }  
  
 for(int i=0;i<mix.length;i++){  
 arr[s+i]=mix[i];  
 }  
}

public class Main {  
  
  
 public static void main(String[] args) throws java.lang.Exception{  
 Sorting b=new Sorting();  
 int[] arr={3,5,2,1,4};  
 b.Cyclic(arr);  
 System.*out*.println(Arrays.*toString*(arr));  
  
 }  
  
}