Linear Search

import java.util.Scanner;

class LinearSearch{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int array[]=new int[size];

System.out.println("Enter array elements:");

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

System.out.print("Element " + (i + 1) + ": ");

array[i] = scanner.nextInt();

}

System.out.print("Enter the value to search for: ");

int target = scanner.nextInt();

int result = linearSearch(array, target);

// Displaying the result

if (result == -1) {

System.out.println("Element not found in the array.");

}

else {

System.out.println("Element found at index: " + result);

}

}

public static int linearSearch(int[] array, int target) {

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

{

if (array[i] == target) {

return i;

}

}

return -1;

}

}

Binary Search

import java.util.Scanner;

class BinarySearch{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int array[]=new int[size];

System.out.println("Enter array elements:");

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

System.out.print("Element " + (i + 1) + ": ");

array[i] = scanner.nextInt();

}

System.out.print("Enter the value to search for: ");

int target = scanner.nextInt();

int result = binarySearch(array, target,0,size-1);

if (result == -1) {

System.out.println("Element not found in the array.");

}

else {

System.out.println("Element found at index: " + result);

}

}

public static int binarySearch(int[] array, int target,int start,int end) {

while(start<=end){

int mid=(start+end)/2;

if(array[mid]==target)

return mid;

else if(array[mid]>target){

end=mid-1;

}

else{

start=mid+1;

}

}

return -1;

}

}

Bubble Sort

import java.util.Scanner;

class BubbleSort{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int array[]=new int[size];

System.out.println("Enter elements to sort:");

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

System.out.print("Element "+(i+1)+": ");

array[i]=scanner.nextInt();

}

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

printArray(array);

bubbleSort(array);

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

printArray(array);

}

public static void printArray(int array[]){

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

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

}

System.out.println();

}

public static void bubbleSort(int array[]){

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

int swapped=0;

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

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

int temp=array[j];

array[j]=array[j+1];

array[j+1]=temp;

swapped=1;

}

}

if(swapped==0)

break;

}

}

}

Selection sort

import java.util.Scanner;

class SelectionSort{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int array[]=new int[size];

System.out.println("Enter elements to sort:");

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

System.out.print("Element "+(i+1)+": ");

array[i]=scanner.nextInt();

}

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

printArray(array);

selectionSort(array);

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

printArray(array);

}

public static void printArray(int array[]){

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

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

}

System.out.println();

}

public static void selectionSort(int[] arr) {

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

int minindex=i;

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

if(arr[minindex]>arr[j])

minindex=j;

}

if(minindex!=i){

int temp=arr[minindex];

arr[minindex]=arr[i];

arr[i]=temp;

}

}

}

}

Insertion Sort

import java.util.Scanner;

class InsertionSort{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int array[]=new int[size];

System.out.println("Enter elements to sort:");

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

System.out.print("Element "+(i+1)+": ");

array[i]=scanner.nextInt();

}

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

printArray(array);

insertionSort(array);

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

printArray(array);

}

public static void printArray(int array[]){

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

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

}

System.out.println();

}

public static void insertionSort(int[] arr) {

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

int key=arr[i];

int j;

for(j=i-1;j>=0&&arr[j]>key;j--){

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

}

arr[j+1]=key;

}

}

}

Largest element

import java.util.Scanner;

class LargestElement{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

System.out.print("Enter array size:");

int size=scanner.nextInt();

int arr[]=new int[size];

System.out.println("Enter array elements:");

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

arr[i]=scanner.nextInt();

}

int ele=largestElement(arr);

System.out.println("The Largest Element from the Entered array is "+ele);

}

public static int largestElement(int arr[]){

int largestind=0;

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

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

largestind=i;

}

}

return arr[largestind];

}

}

Maximum sub array

class MaximumSubArray{

public static void main(String args[]){

//int[] nums = {-2, 1, -3, 4, -1, 2, 1, -5, 4}; // Example input

int[] nums = {-2, -5, -3, -4, -1, -2, -1, -5, -4};

int result = maxSubArraySum(nums);

System.out.println("Maximum Subarray Sum: " + result);

}

public static int maxSubArraySum(int[] nums){

int sum=0;

int max\_sum=nums[0];

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

sum=sum+nums[i];

if(sum>max\_sum){

max\_sum=sum;

}

if(sum<0)

sum=0;

}

return max\_sum;

}

}

Reverse number

import java.util.Scanner;

class reverse{

public static void main(String args[]){

int reverse=0;

int num;

int digit;

Scanner scanner=new Scanner(System.in);

System.out.print("Enter number:");

num=scanner.nextInt();

while(num!=0){

digit=num%10;

reverse=reverse\*10+digit;

num=num/10;

}

System.out.println("the reverse of entered num is "+reverse);

}

}

Prime or not

Every composite number n can be factored into two factors, say a and b, such that n=a×b.

If both a and b were greater than sqrt(n), then a×b would be greater than n, which contradicts the assumption that n=a×b.

Therefore, at least one of these factors must be less than or equal to sqrt(n).

If n has a factor larger than sqrt(n), then it must also have a corresponding

factor smaller than sqrt(n).

import java.util.Scanner;

class PrimeOrNot{

public static void main(String args[]){

Scanner scan=new Scanner(System.in);

System.out.print("Enter number for checking prime or not:");

int num=scan.nextInt();

System.out.print(primeCheck(num));

}

public static boolean primeCheck(int num){

if(num<=1)

return false;

for(int i=2;i\*i<=num;i++){

if(num%i==0)

return false;

}

return true;

}

}

Missing number finder

/\*if 0 is involved in nums[] then n\*(n+1)/2

if 1 onwards start : nums.len+1 then n\*(n+1)/2

\*/

public class MissingNumberFinder {

public static int findMissingNumber(int[] nums) {

int n = nums.length+1 ;

int totalSum = n \* (n + 1) / 2; // Sum of first n natural numbers

int arraySum = 0;

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

arraySum += nums[i];

}

return totalSum - arraySum;

}

public static void main(String[] args) {

int[] nums = {1, 2, 3, 5};

System.out.println("The missing number is: " + findMissingNumber(nums));

}

}

Rearrange by sign

class RearrangeArrayBySign {

public static void main(String args[]){

int nums[]={3,1,-2,-5,2,-4};

//output : 3 -2 1 -5 2 -4

int ans[]=new int[nums.length];

int pos=0,neg=1;

for(int i:nums){

if(i>0){

ans[pos]=i;

pos+=2;

}

else{

ans[neg]=i;

neg+=2;

}

}

for(int j:ans){

System.out.print(j+" ");

}

}

}

RemoveDuplicatesFromSortedArray

public class RemoveDuplicatesFromSortedArray{

public static void main(String args[]){

int nums[]={1,1,2};

System.out.println(removeDuplicates(nums));

}

public static int removeDuplicates(int[] nums){

int count=0;

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

if(i<nums.length-1 && nums[i]==nums[i+1] ){

continue;

}

else{

nums[count]=nums[i];

count++;

}

}

return count;

}

}

Duplicates in array

import java.util.\*;

class Duplicates{

public static void main(String args[]){

int arr[]={1,2,2,3,3,4,5};

HashSet<Integer> set=new HashSet<>();

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

if(arr[i]==arr[i+1]){

set.add(arr[i+1]);

}

}

System.out.println(set);

}

}

Union and Intersection

import java.util.\*;

class UnionIntersection{

public static void main(String args[]){

int a[]={1,2,3,4,4,5,5};

int b[]={1,2,3,4,4,6,7,7};

HashSet<Integer> set=new HashSet<>();

HashSet<Integer> result=new HashSet<>();

HashSet<Integer> unionres=new HashSet<>();

//Intersection

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

set.add(a[i]);

}

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

if(set.contains(b[i])){

result.add(b[i]);

}

}

//Union

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

unionres.add(a[i]);

}

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

unionres.add(b[i]);

}

System.out.println(set);

System.out.println("Intersection result : "+result);

System.out.println("Union result : "+unionres);

}

}

RotateArray

import java.util.Arrays;

public class RotateArray{

public static void main(String args[]){

int nums[]={1,2,3,4,5,6,7}; //{1,2,3};

int k=3;//k=4; k is greater than len of the array

System.out.print(Arrays.toString(rotate(nums,k)));

}

public static int[] rotate(int[] nums,int k){

int n=nums.length;

//k=k%n;

reverse(nums,0,n-1);

reverse(nums,0,k-1);

reverse(nums,k,n-1);

return nums;

}

public static void reverse(int[] nums,int start,int end){

while(start<=end){

int temp=nums[start];

nums[start]=nums[end];

nums[end]=temp;

start++;

end--;

}

}

}

SearchInRotatedSortedArrays

public class SearchInRotatedSortedArrays{

public static void main(String args[]){

int[] nums={4,5,6,7,0,1,2};

int target=0;

System.out.println(search(nums,target));

}

public static int search(int[] nums,int target){

int start=0;

int end=nums.length-1;

while(start<=end){

int mid=(start+end)/2;

if(target==nums[mid]){

return mid;

}

if(nums[start]<=nums[mid]){

if(target>=nums[start]&&target<nums[mid]){

end=mid-1;

}

else{

start=mid+1;

}

}

else{

if(target>=nums[mid]&&target<nums[end]){

start=mid+1;

}

else{

end=mid-1;

}

}

}

return -1;

}

}

String transformation

class StringTransformation{

public static void main(String args[]){

String str="abaabbcc";

System.out.println("transformed string: "+transformString(str));

}

public static String transformString(String str){

char charArray[]=str.toCharArray();

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

if(charArray[i]=='a')

charArray[i]='b';

else if(charArray[i]=='b')

charArray[i]='a';

}

return new String(charArray);

}

}

Reverse words

class ReverseWords{

public static void main(String args[]){

String str="Hello World";

System.out.println("Reverse of String:");

String reversed="";

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

reversed+=str.charAt(i);

}

System.out.println(reversed);

System.out.println("Reverse of words:");

String words[]=str.split(" ");

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

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

}

}

}

Anagram checker

public class AnagramChecker {

public static boolean isAnagram(String s, String t) {

if (s.length() != t.length()) {

return false;

}

int[] count = new int[26];

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

count[s.charAt(i) - 'a']++;

count[t.charAt(i) - 'a']--;

}

for (int c : count) {

if (c != 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

String s = "listen";

String t = "silent";

boolean result = isAnagram(s, t);

System.out.println("Are \"" + s + "\" and \"" + t + "\" anagrams? " + result);

}

}

Palindrome or not

public class PalindromeChecker{

public static void main(String args[]){

//String str="malayalam";

String str="apple";

int flag=0;

int left=0;

int right=str.length()-1;

while(left<right){

if(str.charAt(left)!=str.charAt(right))

flag=1;

left++;

right--;

}

if(flag==1){

System.out.println("Not Palindrome");

}

else{

System.out.println("Palindrome");

}

}

}

arrayStringsAreEqual

class Solution {

public boolean arrayStringsAreEqual(String[] word1, String[] word2) {

StringBuilder string1 =new StringBuilder();

StringBuilder string2 =new StringBuilder();

for(int i=0;i<word1.length;i++){

string1.append(word1[i]);

}

for(int i=0;i<word2.length;i++){

string2.append(word2[i]);

}

if(string1.toString().equals(string2.toString()))

return true;

else

return false;

}

}

ElementFrequency

public class ElementFrequency {

public static void main(String[] args) {

int[] array = {1, 2, 2, 3, 4, 4, 4, 5};

HashMap<integer, Integer> frequencyMap = new HashMap<>();

for (int element: array) {

if (frequencyMap.containsKey(element)) {

frequencyMap.put(element, frequencyMap.get(element) + 1);

}

else { frequencyMap.put(element, 1);

}

for (int key: frequencyMap.keySet()) {

System.out.println("Element:"+key+" Frequency:" + frequencyMap.get(key));

}

}}