

PLACEMENT PREPRATION

Praveen kumar chandaliya

WWW.JAVAPADHO.COM SDJ INFOSOFT PVT LTD 86/26 Pratap Nagar,Jaipur

Program 1: is Unique: implement an algorithm to determine if a string has all unique characters. (Note: use any additional data structure.)

```
public class Unique
{
    static boolean isUniqueChars(String str)
    {
        if(str.length()>128)
        {
            return false;
        }
        boolean [] char_set = new boolean[128];

        for(int i=0;i<str.length();i++)
        {
            int val = str.charAt(i);
            System.out.println(val);
            if(char_set[val])
            {
                return false;
            }
            char_set[val] = true;
        }
        return true;
    }
    public static void main(String[] args)
    {
        String[] words = {"abcde", "hello", "apple", "kite", "padle"};
        for (String word : words) {
            System.out.println(word + ": " + isUniqueChars(word));
        }
    }
}
```

Output:
abcde: true
hello: false
apple: false
kite: true
padle: true

Program 2: Check Permutation: Given two string, write a method to decide if one is a permutation of the other.

```
public class Permutation {
```

```
public static String sort(String s) {
    char[] content = s.toCharArray();
    java.util.Arrays.sort(content);
    return new String(content);
}

public static boolean permutation(String s, String t) {
    return sort(s).equals(sort(t));
}

public static void main(String[] args) {
    String[][] pairs = {"apple", "papel"}, {"carrot", "tarroc"}, {"hello", "lloh"};
    for (String[] pair : pairs) {
        String word1 = pair[0];
        String word2 = pair[1];
        boolean anagram = permutation(word1, word2);
        System.out.println(word1 + ", " + word2 + ": " + anagram);
    }
}
```

Output :

```
apple, papel: true
carrot, tarroc: true
hello, lloh: false
```

Program 3: Write a method to replace all spaces in string with '%20' you may assume that the string has sufficient space at the end to hold the additional characters and that you are given the "true" length of the string.

Example:

```
"www javapadho com "
"www%20javapadho%20com"
```

```
public class URLify {
    // Assume string has sufficient free space at the end
    public static void replaceSpaces(char[] str, int trueLength) {
        int spaceCount = 0, index, i = 0;
        for (i = 0; i < trueLength; i++) {
            if (str[i] == ' ') {
                spaceCount++;
            }
        }
        index = trueLength + spaceCount * 2;
        if (trueLength < str.length)
            str[trueLength] = '\0';
        for (i = trueLength - 1; i >= 0; i--) {
```

```
        if (str[i] == ' ') {
            str[index - 1] = '0';
            str[index - 2] = '2';
            str[index - 3] = '%';
            index = index - 3;
        } else {
            str[index - 1] = str[i];
            index--;
        }
    }
}

public static int findLastCharacter(char[] str) {
    for (int i = str.length - 1; i >= 0; i--) {
        if (str[i] != ' ') {
            return i;
        }
    }
    return -1;
}

public static String charArrayToString(char[] array) {
    StringBuilder buffer = new StringBuilder(array.length);
    for (char c : array) {
        if (c == 0) {
            break;
        }
        buffer.append(c);
    }
    return buffer.toString();
}

public static void main(String[] args) {
    String str = "www javapadho com ";
    char[] arr = str.toCharArray();
    int trueLength = findLastCharacter(arr) + 1;
    replaceSpaces(arr, trueLength);
    System.out.println("\n" + charArrayToString(arr) + "\n");
}
}
```

Object 1: Write a program to check whether 2 strings given by the user are anagram strings or not.

For example:

str1: Are you engineer.

str2: You are engineer.

Output: yes

str1: Am i fine.

str2: I'm fine.

Output: No

str1: Am i fine.

str2: I am fire.

Output: No

```
public class AnagramSecond {
    public static void main(String args[])
    {
        int i=0;
        int h=0;
        String str1="Are you engineer";
        String str2="You are engineer";
        String str[]=new String[3] ;
        String str3[]=new String[3];

        StringTokenizer token = new StringTokenizer(str1," ");
        StringTokenizer token1=new StringTokenizer(str2," ");
        while(token.hasMoreTokens())
        {
            str[i]=token.nextToken();
            i++;
        }
        for(int j=0;j<str.length;j++)
        {
            System.out.println(str[j]);
        }
        while(token1.hasMoreTokens())
        {
            str3[h]=token1.nextToken();
            h++;
        }
        for(int j=0;j<str3.length;j++)
        {
            System.out.println(str3[j]);
        }
        int count=0;
        for( i=0;i<str.length;i++)
        {
            for(h=0;h<str3.length;h++)
            {
                if(str[i].equalsIgnoreCase(str3[h]))
                {
```

```
        count++;
        //break;
    }

    }
    //count++;
}
if(count==str.length)
{
    System.out.println("anagram");
}
else
    System.out.println("sorry");
}
```

```
}
```

Second Solution Using Collection

class Anagram

```
{
```

```
    static void check(String str1,String str2)
```

```
    {
```

```
        String str1all[] = str1.split(" ");
```

```
        String str2all[] = str2.split(" ");
```

```
        ArrayList<String> liststr1 = new ArrayList<String>();
```

```
        ArrayList<String> liststr2 = new ArrayList<String>();
```

```
        for(int i=0;i<str1all.length;i++)
```

```
        {
```

```
            liststr1.add(str1all[i]);
```

```
        }
```

```
        for(int i=0;i<str2all.length;i++)
```

```
        {
```

```
            liststr2.add(str2all[i]);
```

```
        }
```

```
        System.out.println(liststr1);
```

```
        System.out.println(liststr2);
```

```
        boolean flag = liststr1.containsAll(liststr2);
```

```
        if(flag==true)
```

```
        {
```

```
            System.out.println("String first and Second are Anagram "+flag);
```

```
        }
        else
        {
            System.out.println("String first and Second are notAnagram "+flag);
        }
    }
}
public class AnagramProgram
{
    public static void main(String[] args)throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br =new BufferedReader(isr);
        System.out.println("enter first String ");
        String str1 = br.readLine();
        System.out.println("enter second String ");
        String str2 = br.readLine();

        Anagram.check(str1.toLowerCase(),str2.toLowerCase());
    }
}
```

Object 2: Write a program to get the first non repeating alphabet from the given string by the user

For example:

string = abcba

Output : c

string = abcdecbae

Output : d

string =naveen

Output : a

```
package com.sdj;
```

```
class Demo
```

```
{
    static void check(String str)
    {
        char all[]= str.toCharArray();
        int count=0;
        int flag=1;
        for(int i=0;i<all.length;i++)
        {
            count=0;
            char ch = all[i];
            for(int j=0;j<all.length;j++)
            {
```

```
        if(all[j]==ch)
        {
            count++;
        }
    }
    if(count==flag)
    {
        System.out.println("Result"+ch);
        break;
    }
}

}

}
}
public class RepatingAlphabet
{
    public static void main(String[] args) throws IOException
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br =new BufferedReader(isr);
        System.out.println("enter String ");
        String str = br.readLine();
        Demo.check(str);
    }
}
```

Object3:SeatPlaning

Write a function for seat allocate and seat reserved. Seat allocate array and seat reserve array.Seatallocate array is of 10*20 and each row and column represent A1,A2.....;B1,B2.....;.....J1,J2... and so on i.e. row are A to J whereas col starts from 0 to 19.Each cell in the table represent either 0 or1. 0 rep seat available , 1 repr seat reserved. Seat allocation starts from highest to lowest. And row j is highest, i is second highest and so on.Max 20 seats can be booked at a time. if seat is available print the seat no like "B2" i.e (2 row, 3 col)and seat is booked." otherwise Print "Seat is not available."

```
package com.sdj;
public class Seat_Res {
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter dimension of array");
        int r=sc.nextInt();
        int c=sc.nextInt();
        int[][] ar=new int[r][c];
```

```
for(int i=0;i<r;i++)
{
    for(int j=0;j<c;j++)
    {
        ar[i][j]=sc.nextInt();
    }
}
String ans="";
do
{
    System.out.println("enter seat location to book");
    System.out.println("row=");
    int m=sc.nextInt();
    System.out.println("column=");
    int n=sc.nextInt();
    if(ar[m-1][n-1]==0)
    {
        int num=m-1+65;
        char ch=(char)num;
        System.out.println("seat no="+ch+n);
        ar[m-1][n-1]=1;
    }
    else
    {
        System.out.println("seat not available");
    }
    System.out.println("do you want to book more seat(y/n)");
    ans=sc.next();
    if(ans.equalsIgnoreCase("n"))
    {
        System.out.println("thank you");
        break;
    }
}while(ans.equalsIgnoreCase("y"));

}

}
```

Object 4: A string of character is given. Find the highest occurrence of a character and display that character.
eg.: INPUT: AEGBCNAVNEETGUPTAEDAGPE
OUTPUT:

```
E
or
I,(if equal occurrence)
package com.sdj;
public class HighestOccur
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the String");
        String str = sc.nextLine();
        int aplha[] = new int[26];
        char ch;
        int count=0;
        for(int i=0;i<str.length();i++)
        {
            ch =str.toUpperCase().charAt(i);
            int ascii = (int)ch-65;
            count=aplha[ascii];
            aplha[ascii]=++count;
        }
        int value = aplha[0];
        int indexValue = 0;
        for(int i=0;i<aplha.length;i++)
        {
            if(value<aplha[i])
            {
                value = aplha[i];
                indexValue = i;
            }
        }
        System.out.println((char)(indexValue+65));
    }
}
```

Object 5: A mXn matrix was given and rows and column were sorted as shown below then we had to/write a function that search a desired entered no in the matrix .with minimum complexity

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

```
package com.sdj;
```

```
import java.util.Scanner;

public class TwoDSort
{
public static void main(String[] args)
{
int x[][]={{1,2,3,4},
           {5,6,7,8},
           {9,10,11,12},
           {13,14,15,16}};

int i,j,mid=-1,item;
int row=-1;
int low,high,count=0;
Scanner sc=new Scanner(System.in);
System.out.println("enter the number to be searched");
item=sc.nextInt();

for(i=0;i<4;i++)
{
    if(x[i][0]<=item && item<=x[i][3])
    {
        row=i;
        break;
    }
}

low=0;
high=4;
while(low<high)
{
    mid=low+high/2;
    if(x[row][mid]==item)
    {
        count++;
        break;
    }
    else if(x[row][mid]>item)
    {
        count++;
        high=mid-1;
    }
    else
    {
        count++;
    }
}
```

```
        low=mid+1;
    }
}
System.out.println("element found at"+row+","+mid);
System.out.println("count"+count);
}
}
```

Object 6: find the all possible combination of digits ranging 1 to 9 whose sum is 10, no digit shud be repeated in any combination.

Example

1234
127
136
145
19
235
28
37
46

```
package com.sdj;
```

```
public class SumOfDigitOneToTen
{
    public static void main(String[] args) {

        int i,j,k,l,sum;
        for(i = 1;i<10;i++)
        {
            sum = 0;
            for ( j=i+1 ;j<10; j++)
            {
                sum = i+j;
                if(sum == 10)
                {
                    System.out.println(""+i +""+j);
                    break;
                }
            }
            if(sum > 10)
                break;
            for(k=j+1;k<10;k++)
            {
                sum = i+j+k;
                if(sum == 10)
                {
```

```
        System.out.println(i+""+j+""+k);
        break;
    }
    if(sum > 10)
        break;

    for ( l=k+1;l<10;l++)
    {
        sum = i+j+k+l ;
        if(sum == 10)
        {
            System.out.println(i+""+j+""+k+""+l);
            break;
        }
    }
} } } } }
```

Object 7 :

Write a program to find out the combination of an element of each array gives a result 0.

For example:

array 1: {2,1,4,7}

array 1: {3,-3,-8,0}

array 1: {-1,-4,-7,6}

Output:

[2,-8,6]

[1,3,-4]

[1,0,-1]

[4,3,-7]

[4,-3,-1]

[4,0,-4]

[7,-3,-4]

[7,0,-7]

```
package com.sdj;
public class ArraySumZero
{
    public static void main(String[] args)
    {
        int arr1[]={2,1,4,7};
        int arr2[]={3,-3,-8,0};
        int arr3[]={-1,-4,-7,6};
        for(int i=0;i<4;i++)
        {
            for(int j=0;j<4;j++)
```

```
        {
            for(int k=0;k<4;k++)
            {
                if(arr1[i]+arr2[j]+arr3[k]==0)
                    System.out.println(""+arr1[i]+","+arr2[j]+","+arr3[k]+"");
            }
        }
    }
}
```

Object 8:

There is a array of 99 cells and we have to enter 1-100 elements in it , no two elements would repeat , so the is one no. missing because 99 cells and 1-100 nos. so we had to

Implement a function to find that missing no.

```
package com.sdj;
import java.util.Scanner;
public class RandomSum
{
    public static void main(String[] args)
    {
        int all[] = new int[99];
        int sum=0;
        Scanner sc = new Scanner(System.in);
        for(int i=0;i<all.length;i++)
        {
            all[i] =sc.nextInt();
        }

        for(int i=0;i<all.length;i++)
        {
            System.out.print(all[i]+"\\t");
        }
        sum = (all.length*(all.length+1))/2;
        System.out.println(sum);

        for(int i=0;i<all.length;i++)
        {
            sum = sum-all[i];
        }
        System.out.println("Number Not Present is Sum"+sum);
    }
}
```

Zero Matrix: Write an algorithm such that if an element in an M*N matrix is 0, its entire row and column are set to 0.

```
public class ZeroMatrix
{
    public static void nullifyRow(int[][] matrix, int row) {
        for (int j = 0; j < matrix[0].length; j++) {
            matrix[row][j] = 0;
        }
    }

    public static void nullifyColumn(int[][] matrix, int col) {
        for (int i = 0; i < matrix.length; i++) {
            matrix[i][col] = 0;
        }
    }

    public static void setZeros(int[][] matrix) {
        boolean[] row = new boolean[matrix.length];
        boolean[] column = new boolean[matrix[0].length];

        // Store the row and column index with value 0
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                if (matrix[i][j] == 0) {
                    row[i] = true;
                    column[j] = true;
                }
            }
        }

        // Nullify rows
        for (int i = 0; i < row.length; i++) {
            if (row[i]) {
                nullifyRow(matrix, i);
            }
        }

        // Nullify columns
        for (int j = 0; j < column.length; j++) {
            if (column[j]) {
                nullifyColumn(matrix, j);
            }
        }
    }
}
```

```
}

public static void printMatrix(int[][] matrix) {
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            if (matrix[i][j] < 10 && matrix[i][j] > -10) {
                System.out.print(" ");
            }
            if (matrix[i][j] < 100 && matrix[i][j] > -100) {
                System.out.print(" ");
            }
            if (matrix[i][j] >= 0) {
                System.out.print(" ");
            }
            System.out.print(" " + matrix[i][j]);
        }
        System.out.println();
    }
}

public static int[][] randomMatrix(int M, int N, int min, int max) {
    int[][] matrix = new int[M][N];
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            matrix[i][j] = randomIntInRange(min,max);
        }
    }
    return matrix;
}

public static int randomIntInRange(int min, int max) {
    return randomInt(max + 1 - min) + min;
}

public static int randomInt(int n) {
    return (int) (Math.random() * n);
}

public static void main(String[] args) {
    int nrows = 10;
    int ncols = 15;
    int[][] matrix = randomMatrix(nrows, ncols, -10, 10);
    printMatrix(matrix);

    setZeros(matrix);

    System.out.println();
}
```

```
        printMatrix(matrix);
    }
}
```

Program: Print one time even and one time odd number print using Interthread communication.

```
class MyEven extends Thread
{
    MyOdd b;
    public MyEven(MyOdd b)
    {
        this.b = b;
    }
    public void run()
    {
        synchronized (b)
        {
            for(int i=0;i<=100;i+=2)
            {
                System.out.println(getName()+" :"+i);
                try
                {
                    b.wait();
                    sleep(100);
                }
                catch(InterruptedException ex)
                {
                    ex.printStackTrace();
                }
                b.notifyAll();
            }
        }
    }
}

class MyOdd extends Thread
{
    public void run()
    {
        synchronized (this)
        {
            for(int i=1;i<=100;i+=2)
            {
```

```
        System.out.println(getName()+" :"+i);
        notifyAll();
        try
        {
            wait();
        }
        catch(InterruptedException ex)
        {
            ex.printStackTrace();
        }
    }
}

public class EvenOddDemo
{
    public static void main(String[] args)
    {
        MyOdd odd = new MyOdd();
        MyEven even = new MyEven(odd);
        even.start();
        odd.start();
    }
}
```