

Patters Programs

```
1)
*
* *
* *
* *
*****
```

```
-----
import java.lang.*;
import java.util.*;

class p1 {

    public static void main(String args[]){

        int i;
        int j;

        for(i = 1; i <= 5; i++){

            for(j = i; j <= 5; j++){

                System.out.print(" ");

            }

            for(j = 1; j <= 9; j++){

                if( j == i || i == 5 || j == 1 ){

                    System.out.print(" * ");

                }
                else{

                    System.out.print("  ");

                }

            }

            System.out.println();

        }
    }
}
```

```
-----
2)
*****
* *
* *
* *
*
```

```
import java.lang.*;
import java.util.*;
```

```

class p2{

    public static void main(String args[]){

        for(int i = 1; i <= 5; i++){

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

                System.out.print(" ");

            }

            for(int j = 1; j <= 9; j++){

                if(i == 1 || j == 1 || i+j == 6){

                    System.out.print(" * ");

                }

                else{

                    System.out.print("  ");

                }

            }

            System.out.println();

        }

    }
}

```

```

-----
3)
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```

```

import java.lang.*;
import java.util.*;

```

```

class p3{

    public static void main(String args[]){

        int count = 1;
        for(int i = 1; i <= 5; i++){

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

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

            }
            System.out.println();

        }

    }
}

```

```
}  
}
```

4)

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1
```

```
import java.lang.*;  
import java.util.*;
```

```
class p4{
```

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

```
        int num = 1;  
        int x = 5;
```

```
        for(int i = 0; i < 5; i++){
```

```
            for(int j = 1; j <=x; j++){
```

```
                System.out.print(" ");  
            }
```

```
            num = 1;
```

```
            for(int j = 0; j <= i; j++){
```

```
                System.out.print(num+" ");  
                num = num *(i-j)/(j+1);  
            }  
            x--;  
            System.out.println();
```

```
        }  
    }  
}
```

5)

```
1  
2 1 2  
3 2 1 2 3  
4 3 2 1 2 3 4  
3 2 1 2 3  
2 1 2  
1
```

```
import java.lang.*;  
import java.util.*;
```

```
class p5{
```

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

```
    for(int i = 1; i < 4; i++){
```

```
        for(int j = i; j <= 4; j++){
```

```
            System.out.print(" ");
        }
```

```
        for(int j = i; j > 1; j--){
```

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

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

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

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

```
    int x = 5;
```

```
    for(int i = 1; i <= 4; i++){
```

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

```
            System.out.print(" ");
        }
```

```
        x--;
```

```
        for(int j = x; j > 1; j--){
```

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

```
        for(int j = 1; j <= x; j++){
```

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

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

```
}
```

```
}
```

```
6)
```

```
1
```

```
2 1
```

```
3 2 1
```

```
4 3 2 1
```

```
5 4 3 2 1
```

```
import java.lang.*;
```

```
import java.util.*;

class p6{

    public static void main(String args[]){

        for(int i = 1; i <= 5; i++){

            for(int j = i; j >= 1; j--){

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

            }

            System.out.println();

        }

    }

}
```

```
-----
7)
10101
01010
10101
01010
10101
```

```
import java.lang.*;
import java.util.*;

class p7{

    public static void main(String args[]){

        for(int i = 1; i <= 5; i++){

            if(i % 2 != 0){

                for(int j = 1; j <= 5; j++){

                    if(j % 2 != 0){

                        System.out.print("1 ");

                    }

                    else{

                        System.out.print("0 ");

                    }

                }

            }

            else{

                for(int j = 1; j <= 5; j++){

                    if(j % 2 != 0){
```

```

        System.out.print("0 ");
    }

    else{

        System.out.print("1 ");
    }
}

System.out.println();
}
}
}

```

8)
1
10
101
1010
10101

```

import java.lang.*;
import java.util.*;

class p8{

    public static void main(String args[]){

        for(int i = 1; i <= 5; i++){

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

                if(j == 2 || j == 4){

                    System.out.print("0 ");
                }

                else{

                    System.out.print("1 ");
                }
            }

            System.out.println();
        }
    }
}

```

9)
1 2 3 4 5
2 3 4 5

3 4 5
4 5
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5

```
import java.lang.*;
import java.util.*;

class p9{

    public static void main(String args[]){

        int x = 6;
        for(int i = 1; i < 5; i++){

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

                System.out.print(" ");
            }

            for(int j = i; j <= 5; j++){

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

            System.out.println();
        }

        for(int i = 1; i <= 5; i++){

            for(int j = i; j <= 5; j++){

                System.out.print(" ");
            }

            x--;

            for(int j = x; j <= 5; j++){

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

            System.out.println();
        }

    }
}
```

```
B B
C C C
D D D D
E E E E E
F F F F F F
```

```
import java.lang.*;
import java.util.*;

class p10{

    public static void main(String args[]){

        int alpha = 65;
        for(int i = 1; i <= 6; i++){

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

                System.out.print((char)alpha+" ");
            }

            System.out.println();
            alpha++;

        }
    }
}
```

```
11)
A B C D E F
A B C D E
A B C D
A B C
A B
A
A
A B
A B C
A B C D
A B C D E
A B C D E F
```

```
import java.lang.*;
import java.util.*;

class p11{

    public static void main(String args[]){

        for(int i = 1; i <= 6; i++){

            int alpha = 65;
            for(int j = i; j <= 6; j++){
```

12)

A

A B

A B C

A B C D

A B C D E

A B C D E F

```
import java.lang.*;
```

```
import java.util.*;
```

```
class p12{
```

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

```
for(int i = 1; i<= 6; i++){
```

```
for(int j = i; j <= 6; j++){
```

```
System.out.print(" ");
}
```

```
int alpha = 65;
```

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

```
System.out.print((char)alpha+" ");
alpha++;
}
```

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

3

$$\}$$

}

13)
A
B B
C C
D D
E E
F F
E E
D D
C C
B B
A

```
import java.lang.*;
import java.util.*;

class p13{

    public static void main(String args[]){

        int alpha = 65;
        for(int i = 1; i < 6; i++){

            for(int j = i; j <= 6; j++){

                System.out.print(" ");
            }

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

                if(j == i || j == 1){

                    System.out.print((char)alpha+" ");
                }

                else{

                    System.out.print(" ");
                }

            }
            alpha++;
            System.out.println();
        }

        int bet = 70;

        for(int i = 1; i <= 6; i++){

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

                System.out.print(" ");
            }

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

                System.out.print(" ");
            }

            System.out.println();
        }

    }

}
```

```

for(int j = i; j <= 6; j++){

    if(j == i || j == 6 ){

        System.out.print((char)bet+" ");
    }

    else{

        System.out.print(" ");
    }
}
bet--;
System.out.println();
}
}
}

```

```

15)
1 2 3 4 5 6
2 3 4 5 6
3 4 5 6
4 5 6
5 6
6

```

```

import java.lang.*;
import java.util.*;

class p15{

    public static void main(String args[]){

        for(int i = 1; i <= 6; i++){

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

                System.out.print(" ");
            }

            for(int j = i; j <= 6; j++){

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

            System.out.println();
        }
    }
}

```

16) Printing Christmas Tree Using Pyramid

```

import java.lang.*;
import java.util.*;

```

```
class p16{

    public static void main(String[] args){

        int h = 5;
        int w = 5;

        int space = w*5;

        int x = 1;

        for(int a = 1; a <= h; a++){

            for(int i = x; i <= w; i++){

                for(int j = space; j >= i; j--){

                    System.out.print(" ");

                }

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

                    System.out.print("* ");

                }

                System.out.println();

            }

            x = x+2;

            w = w+2;

        }

        for(int i = 1; i <= 4; i++){

            for(int j = space-3; j >= 1; j--){

                System.out.print(" ");

            }

            for(int k= 1; k <= 4; k++){

                System.out.print("* ");

            }

            System.out.println();

        }

    }

}
```

```
17)
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

```
import java.lang.*;
import java.util.*;
```

```
class p17{

    public static void main(String args[]){

        for(int i = 1; i < 5; i++){

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

                System.out.print(" * ");

            }

            System.out.println();

        }

        for(int i = 1; i <= 5; i++){

            for(int j = i; j <= 5; j++){

                System.out.print(" * ");

            }

            System.out.println();

        }

    }
}
```

```
18)
7
7 6
7 6 5
7 6 5 4
7 6 5 4 3
7 6 5 4 3 2
7 6 5 4 3 2 1
```

```
import java.lang.*;
import java.util.*;
```

```
class p18{
```

```

public static void main(String args[]){

    for(int i = 1; i <= 7; i++){
        int x = 7;
        for(int j = 1; j <= i; j++){

            System.out.print(x+" ");
            x--;
        }

        System.out.println();
    }
}

```

```

19)
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
1 2 3 4 5 6 7 6 5 4 3 2 1

```

```

import java.lang.*;
import java.util.*;

class p19{

    public static void main(String args[]){

        for(int i = 1; i <= 7; i++){

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

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

            for(int j = i; j >= 1; j--){

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

            System.out.println();
        }
    }
}

```

```

20)
1234567
234567
34567
4567
567
67

```

7
67
567
4567
34567
234567
1234567

```
import java.lang.*;
import java.util.*;

class p20{

    public static void main(String args[]){

        for(int i = 1; i < 7; i++){

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

                System.out.print(" ");
            }

            for(int j = i; j <= 7; j++){

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

            System.out.println();
        }
        int x = 7;
        for(int i = 1; i <= 7; i++){

            for(int j = i; j <= 7; j++){

                System.out.print(" ");
            }

            for(int j = x; j <= 7; j++){

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

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

21)
1
10
101
1010
10101
101010

1010101

```
import java.lang.*;
import java.util.*;

class p21 {

    public static void main(String args[]){

        for(int i = 1; i <= 7; i++){

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

                if(j == 2 || j == 4 || j == 6){

                    System.out.print("0 ");

                }

                else{

                    System.out.print("1 ");

                }

            }

            System.out.println();

        }

    }

}
```

22)

1111111
1111122
1111333
1114444
1155555
1666666
7777777

```
import java.lang.*;
import java.util.*;

class p22 {

    public static void main(String args[]){

        for(int i = 1; i <= 7; i++){

            for(int j = i; j < 7; j++){

                System.out.print("1 ");

            }

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



```
        System.out.print(i+" ");

    }

    System.out.println();

}
}
}
```

```
23)
1010101
0101010
1010101
0101010
1010101
0101010
1010101
```

```
import java.lang.*;
import java.util.*;
```

```
class p23{

    public static void main(String args[]){

        for(int i = 1; i <= 7; i++){

            if(i % 2 != 0){

                for(int j = 1; j <= 7; j++){

                    if(j % 2 != 0){

                        System.out.print("1 ");
                    }

                    else{

                        System.out.print("0 ");
                    }
                }
            }

            else{

                for(int j = 1; j <= 7; j++){

                    if(j % 2 != 0){

                        System.out.print("0 ");
                    }

                    else{
```

```

        System.out.print("1 ");
    }
}

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

```

24)
 1
 2 6
 3 7 10
 4 8 11 13
 5 9 12 14 15

```

import java.lang.*;
import java.util.*;

class p24{

    public static void main(String args[]){

        for(int i = 1; i <= 5; i++){

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

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

                if(j == 2){

                    System.out.print((i+4)+" ");
                }

                if(j == 3){

                    System.out.print((i+7)+" ");
                }

                if(j == 4){

                    System.out.print((i+9)+" ");
                }

                if(j == 5){

                    System.out.print((i+10)+" ");
                }

            }

            System.out.println();
        }
    }
}

```

```
}
```

```
-----  
25)  
1 2 3 4 5 6 7  
2 3 4 5 6 7 1  
3 4 5 6 7 1 2  
4 5 6 7 1 2 3  
5 6 7 1 2 3 4  
6 7 1 2 3 4 5  
7 1 2 3 4 5 6
```

```
import java.lang.*;  
import java.util.*;
```

```
class p25{  
  
    public static void main(String args[]){  
  
        for(int i = 1; i <= 7; i++){  
  
            for(int j = i; j <= 7; j++){  
  
                System.out.print(j+" ");  
            }  
  
            for(int j = 1; j < i; j++){  
  
                System.out.print(j+" ");  
            }  
  
            System.out.println();  
        }  
    }  
}
```

```
-----  
Class, Objects, Variables, Methods & Constructors Coding Question
```

```
=====
```

```
//Q1. Java Program to Illustrate Use of All Features of Abstract Class
```

```
-----  
abstract class Operations  
{  
    float a = 12, b = 6, c;  
  
    abstract void add();  
    void subtract()  
    {  
        c = a - b;  
        System.out.println("Substraction is : "+c);  
    }  
  
    abstract void multiply();  
    void divide()  
    {
```

```

        c = a / b;
        System.out.println("Division is : "+c);
    }
}
public class Demoabs extends Operations
{
    void add()
    {
        c = a + b;
        System.out.println("Result:"+c);
    }

    void multiply()
    {
        c = a * b;
        System.out.println("Result:"+c);
    }
    public static void main(String[] args)
    {
        Demoabs obj = new Demoabs();
        obj.add();
        obj.subtract();
        obj.multiply();
        obj.divide();
    }
}

```

//Q.2 Java Program to Illustrates Use of Abstract Class and Method

```

abstract class Main
{
    abstract void show();
    public void display()
    {
        System.out.println("This is Java Programming");
    }
}

class Ques2 extends Main
{
    void show()
    {
        System.out.println("Hello world");
    }
    public static void main(String[] args) {

        Ques2 obj = new Ques2();
        obj.show();
        obj.display();
    }
}

```

//Q3. Java Program to Illustrates Use of Instance Inner Class

```

class Outer
{
    void display()
    {
        System.out.println("this is Outer class");
        Inner obj1=new Inner();
        obj1.show();
    }
}
class Inner
{
    void show()
    {
        System.out.println("This is Inner class");
    }
}

```

```

class Ques3
{
    public static void main(String[] args)
    {
        Outer obj = new Outer();
        obj.display();
    }
}

```

//Q4. Java Program to Illustrates Use of Static Inner Class

//example 1:-

```

public class Ques4{ //outer class
//int x=100;
static int a=10;

    static class Inner{
        void display()
        {
            System.out.println("this is Innerclass and access msg of the outerclass " + a);
        }

        void display1(){
            System.out.println(a);
        }
    }

    public static void main(String[] args){
        Inner obj = new Inner();

        obj.display();
        System.out.println("Inner class accessed by outer class object");
        Ques4.Inner obj1=new Ques4.Inner();
        obj1.display1();
    }
}

```

```
}  
}
```

//Q5. Java Program to Illustrates Use of Referencing the Object from Inner Class

```
public class Ques5 { //outer class  
  
    void show()  
    {  
        System.out.println("HII");  
    }  
    public static class Inner {  
        void display()  
        {  
            System.out.println("Welcome to the world!! ");  
            Ques5 obj1=new Ques5();  
            obj1.show();  
        }  
    }  
    public static void main(String[] args){  
  
        Ques5.Inner obj = new Ques5.Inner();  
        obj.display();  
    }  
}
```

// Q6. Java Program to Create Outer Class Bank Account and the Inner Class Interest in it

```
class Account {  
  
    int principal=200,rate=4,time=2;  
    void test()  
    {  
        Inner obj=new Inner();  
        obj.showInterest();  
    }  
    class Inner {  
        void showInterest()  
        {  
            int si=(principal*rate*time)/100;  
            System.out.println("Interest is : "+si);  
        }  
    }  
}  
  
public class Ques6 {  
    public static void main(String[] args){  
        Account ac = new Account();  
        // System.out.println("Enter amount = ");  
        //int x=ac.nextInt();  
        ac.test();  
  
    }  
}
```

//Q7. Java Program to Implement Shape Interface using Circle and Rectangle Class

```
interface Shape
{
    void values();
    void result();
}

class Circle implements Shape{
    int r=0,area=0;
    double pi=3.14;
    public void values()
    {
        int r=15;
    }

    public void result()
    {
        area=pi*r*r;
        System.out.println("claculate circle is = "+area);
    }
}

class Rectangle implements Shape{
    int l=0,b=0;
    double rectarea;

    public void values(){
        l=10;
        b=20;
    }
    public void result()
    {
        rectarea= l*b;
        System.out.println("Calculated Rectangle = "+rectarea);
    }
}

public class Ques7{
    public static void main(String[] args){
        Rectangle obj = new Rectangle();
        obj.values();
        obj.result();
    }
}
```

//Q8. Java Program to Count Number of Objects Created for Class

```
public class CountObj{
    static int count;

    CountObj()
    {
        count++;
    }
}
```

```

    }

    public static void main(String[] args)
    {
        CountObj n1 = new CountObj();
        CountObj n2 = new CountObj();
        CountObj n3 = new CountObj();
        CountObj n4 = new CountObj();
        CountObj n5 = new CountObj();

        System.out.println("Number of objects in class = "+count);
    }
}

```

//Q9. Java Program to Implement the Passing and Returning Objects

```

public class Pass_Return
{
    int l=12,b=5,area;

    Pass_Return area1(Pass_Return obj1)
    {
        obj1 = new Pass_Return();
        obj1.l=this.l;
        obj1.b=this.b;
        obj1.area=obj1.l * obj1.b;
        return obj1;
    }

    Pass_Return area2(Pass_Return obj2)
    {
        obj2 = new Pass_Return();
        obj2.l=this.l ;
        obj2.b=this.b ;
        obj2.area=obj2.l * obj2.b;
        return obj2;
    }

    public static void main(String[] args)
    {
        Pass_Return obj = new Pass_Return();

        Pass_Return a = obj.area1(obj);
        Pass_Return b = obj.area1(obj);

        System.out.println("Area1 : "+a.area);
        System.out.println("Area2 : "+b.area);
    }
}

```

//Q10. Java Program to Swap Objects using Swap() Method

```

class Swap_obj{
    int number=0;

```



```

Swap_obj(int number)
{
    this.number=number;
}
}
class Ques10{
public static void swap(Swap_obj s1,Swap_obj s2)
{
    int temp=s1.number;
    s1.number=s2.number;
    s2.number=temp;
}

public static void main(String[] args)
{
    Swap_obj s1 = new Swap_obj(10);
    Swap_obj s2 = new Swap_obj(20);

    System.out.println("before swap "+s1.number+ " , "+s2.number);
    swap(s1,s2);
    System.out.println("After swap");
    System.out.println(s1.number);
    System.out.println(s2.number);

}
}

```

// 11. Java Program to Illustrate Use of Methods in a Class

```

public class Ques11
{
    void showMarks(int m1,int m2,int m3)
    {
        int res=m1+m2+m3;
        System.out.println("Result of Marks in 3 subject is = "+res);
    }

    void add(int a, int b)
    {
        int sum = a+b;
        System.out.println("Sum of two values is = "+sum);
    }

    void multi(int a,int b)
    {
        int res=a*b;
        System.out.println("Multiplication of two number is = "+res);
    }

    public static void main(String[] args)
    {
        Ques11 obj = new Ques11();
        obj.showMarks(15,20,29);
    }
}

```

```
obj.add(20,45);  
obj.multi(45,5);
```

```
}  
}
```

//12. Java Program to Create a Method without Parameters and with Return Type

```
import java.util.*;  
class Main  
{  
    int a = 99,b=56,res=0;  
  
    int displayres()  
    {  
        int res = a+b;  
        return res;  
    }  
}  
public class Ques12  
{  
    public static void main(String[] args)  
    {  
        Main obj = new Main();  
        int result = obj.displayres();  
        System.out.println("Result = "+result);  
    }  
}
```

//Q13. Java Program to Create a Method without Parameters and Return Type

```
import java.util.*;  
class Main  
{  
    int a = 99,b=56,res=0;  
  
    void displayres()  
    {  
        int res = a+b;  
        System.out.println("Result = "+res);  
    }  
}  
public class Ques13  
{  
    public static void main(String[] args)  
    {  
        Main obj = new Main();  
        obj.displayres();  
    }  
}
```

//Q14. Java Program to Create a Method with 2 Parameters and without Return Type

```

import java.util.*;
public class Ques14
{

    public static void displayres(int a,int b)
    {
        int res = a*b;
        System.out.println("Result = "+res);
    }

    public static void main(String[] args)
    {
        Ques14 obj = new Ques14();
        obj.displayres(40,50);
    }
}

```

//Q16. Java Program to Illustrate Use of Final Keyword

```

final class Main{    //final class created

    public final void show() // final method
    {
        System.out.println("This is final Method");
    }
}
class Ques16
{
    /* public final void show()    // it will get error
    {
        System.out.println("hello world");
    }
    */

    public static void main(String[] args)
    {
        Main obj = new Main();
        final int number=15; //final variable
        // number = 20; it well get error ,override the value

        System.out.println("Number = "+number);

        obj.show();
    }
}

```

//Q17. Java Program to Illustrate Use of Constructor.

```

class Main
{
    double length,height;

    Main()
    {
        length = 12.5;
    }
}

```

```

height = 15.2;
}

Main(int a,int b)
{
    length=a;
    height=b;
}

double show()
{
    return length*height;
}
}

public class Ques17
{
    public static void main(String[] args)
    {
        Main obj = new Main();
        double area;
        area = obj.show();
        System.out.println("area = "+area);

        Main obj1=new Main(12,12);
        area=obj1.show();
        System.out.println("area2 = "+area);

    }
}

```

//Q18. Java Program to Illustrates Use of Chaining Constructor

```

class Ques18{

    Ques18()
    {
        this(9);
        System.out.println("Default Constructor");
    }

    Ques18(int a)
    {
        this(9,18);
        System.out.println(a);
        System.out.println("Single value Constructor");
    }

    Ques18(int a,int b)
    {
        System.out.println(a*b);
    }

    public static void main(String[] args)
    {

```

```
Ques18 obj = new Ques18();
```

```
}  
}
```

//Q19. Java Program to Create an Object for Class and Assign Value in the Object using Constructor.

```
public class Ques19  
{  
    String ch;  
    int age;  
  
    Ques19()  
    {  
        System.out.println("Zero value constructor");  
    }  
  
    Ques19(String ch,int age)  
    {  
        this.ch=ch;  
        this.age=age;  
    }  
  
    String getch()  
    {  
        return ch;  
    }  
  
    int getage()  
    {  
        return age;  
    }  
  
    public static void main(String[] args)  
    {  
        Ques19 obj = new Ques19();  
        Ques19 obj1 = new Ques19("Hello",24);  
        System.out.println(obj1.getch());  
        System.out.println(obj1.getage());  
    }  
}
```

//Q20. String Constructor Program in Java

```
public class Ques20  
{  
    public static void main(String[] args)  
    {  
        char[] str={'c','o','m','p','u','t','e','r'};  
        String st="Hello";  
        String st1 = new String("World");  
        String st2 = new String(str);  
        String st3 = new String(str,2,2);  
        System.out.println("st = "+st);  
        System.out.println("st1 = "+st1);  
    }  
}
```

```
System.out.println("st2 = "+st2);
System.out.println("st3 = "+st3);
```

```
}
}
```

//Q21. Java Program to Allocate and Initialize Super Class Members using Constructor.

```
class ConstCall
{
    ConstCall()
    {
        System.out.println("0 aargument constructor");
    }

    ConstCall(int m1,int m2)
    {
        System.out.println("Marks of two subject of two students ");
        int tot = m1 +m2;
        System.out.println(" two subject marks : "+tot);
    }
}

class Ques21 extends ConstCall
{
    Ques21(String name)
    {
        super(15,19);
        System.out.println("Name of student : "+name);
    }

    public static void main(String[] args)
    {
        //Ques21 obj = new Ques21();
        Ques21 obj1 = new Ques21("Harry");

    }
}
```

//Q22. Java Program to Check the Accessibility of Static and Non-Static Variable by a Static Method.

```
class Ques22
{
    static int a=10; //static variable
    int b=30;        //non-static variable

    public static void print(){        //static method
        System.out.println("Static value = "+a);
        // System.out.println(b); it will through error
        // error: non-static variable b cannot be referenced from a static context
    }

    public static void main(String[] args)
    {
        Ques22 obj=new Ques22();
        obj.print();
    }
}
```

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

//Q23. Java Program to Demonstrate Usage of an Instance Variable in the Test Class.

```
public class Test
{
    public int num;
    private String name;
    double amount;

    public Test(int snum)
    {
        num = snum;
    }

    public void setname(String sname)
    {
        name = sname;
    }

    public void setAmt(double samt)
    {
        amount=samt;
    }

    public void printDetails()
    {
        System.out.println("Student name is - "+name+ ". Number is - " +num+ " . Amount is - "+amount);
    }

    public static void main(String[] args)
    {
        Test ts = new Test(101);
        ts.setAmt(2500);
        ts.setname("Harry");
        ts.printDetails();
    }
}
```

//Q24. Java Program to Demonstrate Usage of a Static Variable in the Test Class.

```
class Test1
{
    static int i=20;

    public static void main(String[] args)
    {
        System.out.println(i);
    }
}
```

//Q25. Java Program to Check Whether Which One is Executed First, Static Block or the Static Method.

```

public class Ques25
{
    static int value1=10;
    int value2;

    static {
        System.out.println("This is static Block");
        System.out.println(value1);
    }

    static void show(){
        System.out.println("This is static Method");
    }

    public static void main(String[] args)
    {
        Ques25 obj = new Ques25();
        obj.show();
    }
}

```

//Q26. Java Program to Calculate Sum of Two Byte Values using Type Casting.

```

import java.util.Scanner;
public class SumOfBytes
{
    public static void main(String[] args)
    {
        byte a=10;
        byte b=15;
        int sum;
        sum=a+b;
        System.out.println("Sum of two byte a and b is = "+sum);

    }
}

```

----- Arrays Coding Questions

=====

1. Write a program to print elements of Array ?

```

class Q1 {
    public static void main(String[] args) {

        int [] arr = new int [] {1, 2, 3, 4, 5};
        System.out.println("Elements of given array: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i] + " ");
        }
    }
}

```

2. Write a Java program to check the equality of two arrays?


```

import java.util.Arrays;
class Q2{

    public static void main(String[] args)
    {
        int arr1[] = {10,20,30};
        int arr2[] = {10,20,30};
        boolean result = Arrays.equals(arr1,arr2);

        if(result == true)
        {
            System.out.println("Arrays are equal");
        }
        else
        {
            System.out.println("Arrays are not equal");
        }
    }
}

```

3. Write a Java program to find all pairs of elements in an integer array whose sum is equal to a given number?

```

class Q3 {
    public static void main(String args[])
    {
        int[] arr = { 1, 5, 7, -1, 5 };
        int sum = 6;
        getCount(arr, sum);
    }
    public static void getCount(int[] arr, int sum)
    {

        int count = 0;
        for (int i = 0; i < arr.length; i++)
            for (int j = i + 1; j < arr.length; j++)
                if ((arr[i] + arr[j]) == sum)
                    count++;

        System.out.printf("Count of pairs is %d", count);
    }
}

```

//4. Write a program to reverse an Array in java .

```

import java.util.*;
class Q4{

    public static void main(String[] args)
    {
        int arr[] = new int[] {1,3,5,7,9};
        System.out.println("Print the array");

        for(int i=0;i<arr.length;i++)
        {
            System.out.print(arr[i]+ " ");
        }
    }
}

```

```

    }
    System.out.println();
    System.out.println("=====");
    System.out.println("Reverse array");
    for(int j=arr.length-1;j>=0;j--)
    {

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

```

/*

OUTPUT :

Print the array

1 3 5 7 9

=====

Reverse array

9 7 5 3 1

*/

5. Find out smallest and largest number in a given Array?

```

public class Q5 {

```

```

    public static void main(String[] args) {
        int numbers[] = new int[]{21,43,56,1,81,91,66,23};
        int smallest = numbers[0];
        int largest = numbers[0];

```

```

        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] > largest)
                largest = numbers[i];
            else if (numbers[i] < smallest)
                smallest = numbers[i];
        }

```

```

        System.out.println("Largest Number is : " + largest);
        System.out.println("Smallest Number is : " + smallest);
    }
}

```

//6. Print the third-largest number in an array without sorting it
 //: [24,54,31,16,82,45,67]

```

import java.util.*;

```

```

class Q6

```

```

{
    public static int getThirdLargest(int[]a,int total)

```

```

    {
        int temp;
        for(int i=0;i<total;i++)
        {
            for(int j=i+1;j<total;j++)

```

```

    {
        if(a[i]>a[j])
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}

return a[total-3];
}

public static void main(String[] args)
{
    int a[]={2,5,11,33,7,99};
    System.out.println("Third largest :"+getThirdLargest(a,6));
}
}

/*
OUTPUT:
Third largest :11
*/

```

7. Write a program to merge two arrays of integers by reading one number at a time from each array until one of the array is exhausted, and then concatenating the remaining numbers.

Input: [23,60,94,3,102] and [42,16,74]

Output: [23,42,60,16,94,74,3,102]

```

public class Q7 {
    public static void main(String[] args) {
        int arr1[] = {23,60,94,3,102};
        int n1 = arr1.length;

        int arr2[] = {42,16,74};
        int n2 = arr2.length;

        int arr3[] = new int[n1 + n2];
        mergeArrays(arr1, arr2, n1, n2, arr3);

        System.out.println("Array after merging");
        for (int i=0; i < n1+n2; i++)
            System.out.print(arr3[i] + " ");

    }

    public static void mergeArrays(int[] arr1, int[] arr2, int n1, int n2, int[] arr3){
        int i = 0;
        int j = 0;
        int k = 0;

        while(i < n1){
            arr3[k++] = arr1[i++];
        }
    }
}

```

```

while(j < n2){
arr3[k++] = arr2[j++];
}

}
}

```

9. Write a program which generates the series 1,4,27,16,125,36

```

public class Q9
{
    public static void sequenceSeries(int N)
    {
        for (int i=1;i<=N;i++)
        {
            if (i % 2 == 0)
            {
                System.out.print(i*i);
            }
            else
            {
                System.out.print(i*i*i);
            }
            if (i<N){
                System.out.print(",");
            }
        }
    }

    public static void main(String args[])
    {
        int N = 6;
        sequenceSeries(N);
    }
}

```

11. How to convert a byte array to String?

```

import java.io.IOException;
import java.util.*;

class Q11
{
    public static void main(String[] args) throws IOException
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter any String");
        String str = sc.next();

        byte[] bytes = str.getBytes();

        System.out.println(Arrays.toString(bytes));

        String string = new String(bytes);
    }
}

```

```
}
```

13. Write a program to sort an Array in Java ?

```
public class Q13 {  
    public static void main(String[] args) {  
  
        int [] arr = new int [] {5, 2, 8, 7, 1};  
        int temp = 0;  
        System.out.println("Elements of original array: ");  
        for (int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = i+1; j < arr.length; j++) {  
                if(arr[i] > arr[j]) {  
                    temp = arr[i];  
                    arr[i] = arr[j];  
                    arr[j] = temp;  
                }  
            }  
        }  
  
        System.out.println();  
        System.out.println("Elements of array sorted in ascending order: ");  
        for (int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
    }  
}
```

//14. Write a program to check whether two given Arrays are equal, given both contains same data type and same length ?

```
import java.util.*;  
  
class Q14  
{  
    static boolean Equal(int arr1[],int arr2[])  
    {  
        int a = arr1.length;  
        int b = arr2.length;  
  
        if(a != b)  
            return false;  
  
        Arrays.sort(arr1);  
        Arrays.sort(arr2);  
  
        for(int i=0;i<a;i++)  
  
            if(arr1[i] != arr2[i])  
                return false;  
  
        return true;  
    }  
}
```

```

    }
    public static void main(String[] args)
    {
        int arr1[] = {7,5,9,5,1};
        int arr2[] = {4,7,5,1,2};

        if(Equal(arr1,arr2))
            System.out.println("yes");
        else
            System.out.println("No");
    }
}

```

15. How to find the missing number in a given Array from number 1 to 100 ?

```

import java.util.Scanner;
public class Q15 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the n value: ");
        int n = sc.nextInt();
        int inputA[] = new int[n];
        System.out.println("Enter (n-1) numbers: ");
        for(int i=0; i<=n-2; i++) {
            inputA[i] = sc.nextInt();
        }
        //Finding the missing number
        int sumOfAll = (n*(n+1))/2;
        int sumOfArray = 0;
        for(int i=0; i<=n-2; i++) {
            sumOfArray = sumOfArray+inputA[i];
        }
        int missingNumber = sumOfAll-sumOfArray;
        System.out.println("Missing number is: "+missingNumber);
    }
}

```

//18. How to find duplicate elements in a given Array

```

import java.util.*;

class Q18
{
    public static void main(String[] args)
    {
        int arr[] =new int[]{1,2,3,4,2,7,8,8,3};
        System.out.println("Finding the duplicate element");

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

```

```

        if(arr[i] == arr[j])
        System.out.println(arr[j]);
    }
}
}
}

```

 19. Write a program to sum all the values of a given Array in java?

```

class Q19 {
    public static void main(String[] args){
        int arr[] = { 12, 3, 4, 15 };
        int sum = 0;
        int i;
        for (i = 0; i < arr.length; i++){
            sum += arr[i];
        }
        System.out.println("Sum of given array is "+sum);
    }
}

```

 //How do you separate zeros and non-zeros in a given Array in java?

```

import java.util.*;

class Q20
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size if array : ");
        int size = sc.nextInt();
        int[] arr = new int[size];
        System.out.println("Enter the element in the aarray :");

        for(int i=0;i<size;i++)
        {
            arr[i] = sc.nextInt();
        }

        System.out.println("the array is :"+Arrays.toString(arr));
        System.out.println("resultant array :");

        int temp=0;
        for(int i=0;i<arr.length;i++)
        {
            if(arr[i] != 0)
            {
                arr[temp] = arr[i];
                temp++;
            }
        }
    }
}

```

```

        while(temp<arr.length)
        {
            arr[temp] = 0;
            temp++;
        }
        System.out.println("The array is :"+Arrays.toString(arr));

    }
}

```

21. How to convert Array to ArrayList in java ?

```

import java.util.*;

class Q21
{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter names of students");
        String str = sc.nextLine();

        List names = Arrays.asList(str);

        System.out.println(names);
    }
}

```

//How to convert Array to TreeSet in java ?

```

import java.util.*;

class Q22
{
    public static void main(String[] args)
    {
        Integer[] arr = {5,98,45,4,1,87,2};
        List<Integer> list = Arrays.asList(arr);
        Set<Integer> set = new TreeSet<Integer>(list);
        System.out.println("The set element are :");
        for(Integer var : set)
            System.out.println(var);

    }
}

```

23. How to convert ArrayList to String Array in java ?

```

import java.util.*;

class Q23
{
    public static void main(String[] args)
    {

```



```

ArrayList<String> names = new ArrayList<String>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter names of students (Enter 0 when completed)");

while( sc.hasNext() )
{
    String str = sc.nextLine();
    if( str.isEmpty())
    {
        break;
    }
    names.add(str);
}

String[] str = new String[names.size()];

for (int i = 0; i < names.size(); i++)
{
    str[i] = names.get(i);
}

for (String k : str)
{
    System.out.println(k);
}
}

```

24. Write a program to find second largest element in a given Array in java?

```

import java.util.*;

class SecondLargestElementArray
{
    static int getSecondLarge(int[]arr,int total)
    {
        int temp;
        for(int i=0;i<total;i++)
        {
            for(int j=i+1;j<total;j++)
            {
                if(arr[i]>arr[j])
                {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        return arr[total-2];
    }
    public static void main(String[] args)
    {

```

```

int arr[] = {10,55,66,44,88,22,77,99,33};
System.out.println("Display Second Largest Element :"+getSecondLarge(arr,9));
}
}

```

String Programs

```

import java.lang.String;
import java.util.Map;
import java.util.Scanner;
import java.util.TreeMap;

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

        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        char ch[]=s.toCharArray();
        Map<Character,Integer> m=new TreeMap<>();
        int count=0;
        for (int i = 0; i <s.length(); i++) {
            count=0;
            for (int j = 0; j <s.length() ; j++) {
                if(ch[i]==ch[j]){
                    count++;
                }
            }
            m.put(ch[i],count);
        }
        System.out.println(m);
    }
}

```

5. How to check if a String contains only digits?

```

import java.util.Scanner;

public class digitInString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        char[] ch = str.toCharArray();

        for (int i = 0; i < str.length(); i++) {
            if (Character.isDigit(str.charAt(i)))
                System.out.println("Digit is Present" + str.charAt(i));

            else {
                System.out.println("Not Present");
            }
        }
    }
}

```

```
}  
}
```

1. How to Print duplicate characters from String?

```
import java.util.Scanner;
```

```
public class DuplicateCharacterinString {  
    public static void main(String[] args) {  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Eneter The String = >");  
        String str= sc.nextLine();  
  
        char[] ch=str.toCharArray();  
        System.out.println("Duplicate Characters are == > ");  
        for (int i = 0; i < str.length(); i++) {  
            for (int j = i+1; j <str.length() ; j++) {  
                if(ch[i]==ch[j]){  
                    System.out.println(ch[j]);  
                    break;  
                }  
            }  
        }  
    }  
}
```

4. How to program to print the first non repeated character from String?

```
import java.util.HashMap;  
import java.util.Scanner;
```

```
public class firstNonRepChar {  
    // Using map  
    // public static int NonRepeat(String str){  
    //     HashMap<Character , Integer> map= new HashMap<>();  
    //     for(char ch: str.toCharArray()){  
    //         map.put(ch,map.getOrDefault(ch ,0)+1);  
    //     }  
    //     for (int i = 0; i <str.length() ; i++) {  
    //         char ch=str.charAt(i);  
    //         if(map.get(ch)==1){  
    //             return ;  
    //         }  
    //     }  
    //     return -1;  
    // }  
    // }  
    // public static void main(String[] args) {  
    //     Scanner sc=new Scanner(System.in);  
    //     String str=sc.nextLine();  
    //     System.out.print(NonRepeat(str));  
    // }
```

```

//***** Without map *****
public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter The String ==> ");
    String str=sc.nextLine();

//    System.out.println("Non Repeating Characters in Strings are ==>");
    System.out.println("First Non Repeating Characters are :");
    for (int i = 0; i <str.length() ; i++)
    {
        int temp=0;
        for (int j = 0; j <str.length() ; j++) {

            if(str.charAt(i)==str.charAt(j)){
                temp++;
            }

        }
        if(temp ==1){
            System.out.println(str.charAt(i));
            break;
        }
    }
}

```

19. How to return the highest occurred character in a String?
 For example if input is "aaaaaaaaaaaaaabbcbcddeeeeee" it should return "a".

```

public class highOccuOfCharinString {
    static final int Size=256;

    public static int find(String str) {
        int count[]=new int[Size];
        int count2=0;
        int max=-1;
        char res=' ';
        for (int i = 0; i <str.length() ; i++) {
            count[str.charAt(i)]++;
        }
        for (int i = 0; i <str.length(); i++) {
            if(max<count[str.charAt(i)]){
                max=count[str.charAt(i)];
                res=str.charAt(i);
            }
        }
        for (int i = 0; i <str.length() ; i++) {
            if(str.charAt(i)==res){
                count2++;
            }
        }
        //    char c=(char)count2;
    }
}

```

```

        return str.charAt(count2);
    }

    public static void main(String[] args) {
        String str="SUCCESS";
        System.out.println(find(str));
    }
}

-----

public class numStrToIntStr {

    public static void main(String[] args) {
        String str="9999999999";
        System.out.println("String input = "+Integer.parseInt(str));
        System.out.println("Integer Value = "+Integer.valueOf(str));
    }
}

-----

public class rec {

    public static void display(String str,String res ) {

        if (str.length() == 0) {
            System.out.println(res); //to print empty String
            return;
        }
        for (int i = 0; i < str.length(); i++) {
            char temp = str.charAt(i);
            String op = str.substring(0, i) + str.substring(i + 1);
            display(op, res+temp);
        }

    }

    public static void main(String[] args) {
        String s="ABCD";
        display(s,"");
    }
}

-----

import java.util.Scanner;
public class recursiveFact {
    static String A = "A";
    static String C = "C";
    static String B = "B";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Disk:");
        int n = scanner.nextInt();
        toh(n, A, C, B);
        scanner.close();
    }

    private static void toh(int n, String A, String C, String B) {
        if (n == 1) {

```

```

        System.out.println(A + " --> " + C);
    } else {
        toh(n - 1, A, B, C);
        System.out.println(A + " --> " + C);
        toh(n - 1, B, C, A);
    }
}
}
}

```

15. How to remove duplicate characters from String?

```

import java.util.Scanner;

public class removeDupfromString {

    public static String Duplicate(String str){
        String temp="";
        for (int i = 0; i <str.length() ; i++) {
            char ch=str.charAt(i);
            if(temp.indexOf(ch)== -1){
                temp=temp+ch;
            }
        }
        return temp;
    }
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter Your String");
        String str=sc.nextLine();
        System.out.println("After Removing Duplicate "+Duplicate(str));
    }
}

```

// 11. How to replace each given character to other e.g. blank with %20?

```

import java.util.Scanner;

public class replaceBlank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String with Blank Spaces");
        String str = sc.nextLine();

        System.out.println(str.replaceAll(" ","%20"));
    }
}

```

//14. How to reverse words in a sentence without using a library method?

```

public class reverseWithoutLib {
//    public static void main(String[] args) {
//        String name="Hello There";
//        String rev="";

```

```
//    int len=name.length();
//    for(int i = len-1; i >=0 ; i--) {
//        rev=rev+name.charAt(i);
//    }
//    System.out.println(rev);
// }
```

// Approach 2 Recursive

```
public static String reverse(String str) {
    if ((null == str) || (str.length() <= 1)) {
        return str;
    }
    return reverse(str.substring(1)) + str.charAt(0);
}

public static void main(String[] args) {
    System.out.println(reverse("RAHUL"));
}

}
```

```
import java.util.HashMap;
import java.util.Scanner;
```

```
public class StringAnagrams {
    public static boolean Anagrams(String str1, String str2, int k) {
        HashMap<Character , Integer> map=new HashMap<>();
        if(str1.length()!=str2.length()){
            return false;
        }
        for (int i = 0; i < str1.length() ; i++) {
            char ch=str1.charAt(i);
            map.put(ch,map.getDefault(ch,0) +1);
        }
        for (int i = 0; i <str2.length() ; i++) {
            char ch=str2.charAt(i);
            if(map.getDefault(ch,0) > 0){
                map.put(ch,map.get(ch) -1 );
            }
        }
        int count =0;
        for (char ch:map.keySet()) {
            count +=map.get(ch);
        }
        if(count > k){
            return false;
        }else {
            return true;
        }
    }
}
```

```
public static void main(String[] args) {
    Scanner s=new Scanner(System.in);
```

```
String str1=s.nextLine();
String str2=s.nextLine();
```

```

int k=s.nextInt();
System.out.println(Anagrams(str1,str2,k));
// Take a Input as ABABADDCCC & BBBAACCCED K=3 op=> True
}
}

```

//Q17. How to check if a String is a valid shuffle of two String?

```

public class validShuffelOfTwoStrngs {
    private static void validShuffle(String s1, String s2, String result) {
        String s3 = s1 + s2;
        StringBuffer s = new StringBuffer(s3);
        boolean flag = false;
        char[] ch = result.toCharArray();
        if (s.length() != result.length()) {
            flag = false;
        } else {
            for (int i = 0; i < ch.length; i++) {
                String temp = Character.toString(ch[i]);
                if (s3.contains(temp)) {
                    s.deleteCharAt(s.indexOf(temp));
                    s3 = new String(s);
                    flag = true;
                } else {
                    flag = false;
                    break;
                }
            }
        }

        if (flag) {
            System.out.println("Yes");
        } else {
            System.out.println("No");
        }
    }

    public static void main(String[] args) {

        String s1 = "AABBCCDD";
        String s2 = "ABCDF";

        String result = "AAABBBCCCDDDF";

        validShuffle(s1, s2, result);

    }
}

```


Special Programs

//Q1. Note: a palindrome is a sequence of characters that reads the same forwards and backwards.

Ex: Given the following strings...

"A man, a plan, a canal: Panama.", return true

```
import java.util.*;

class Q1
{
    public static void main(String args[])
    {
        String str1,str2= "";

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the string");

        str1= in.nextLine();

        str1 = str1.toLowerCase().trim().replaceAll("[,;:\s]", "");

        int length = str1.length();

        for ( int i = length - 1; i >= 0; i-- )
        {
            str2 = str2 + str1.charAt(i);
        }
        if (str1.equals(str2))
        {
            System.out.println("The given string is a palindrome.");
        }
        else
        {
            System.out.println("The given string is not a palindrome.");
        }
    }
}
```

//Q2. This question is asked by Google. Given a string, return whether or not it uses capitalization correctly. A string correctly uses capitalization if all letters are capitalized, no letters are capitalized, or only the first letter is capitalized.

Ex: Given the following strings...

"USA", return true

"Calvin", return true

"compUter", return false

"coding", return true

```
import java.util.*;

public class Q2
{
    public static void detectCapitalUse(String word)
    {
        String word1 = word.replaceAll("[,;:\s]", "");
```

```

        if (word1.matches("[A-Z]*|[A-Z][a-z]+|[a-z]+"))
        {
            System.out.println("The given string is properly Capitalized.");
        }
        else
        {
            System.out.println("The given string is not properly Capitalized.");
        }
    }
}

```

```

public static void main (String [] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter your String");

```

```

    String word = sc.nextLine();

```

```

    detectCapitalUse(word);
}
}

```

//Q3. This question is asked by Amazon. Given a string representing the sequence of moves a robot vacuum makes, return whether or not it will return to its original position. The string will only contain L, R, U, and D characters, representing left, right, up, and down respectively.

Ex: Given the following strings...

"LR", return true

"URURD", return false

"RUULLDRD", return true

```

import java.util.*;

```

```

class Q3

```

```

{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Provide instruction to move Robot (U for up, D for down, R for right & L for Left)");

```

```

        String inst = sc.next();

```

```

        Solution sl = new Solution();

```

```

        System.out.println(sl.roboMoves(inst));
    }
}

```

```

class Solution

```

```

{
    public boolean roboMoves(String inst)
    {
        int x = 0;
        int y = 0;

```

```

        char[] array = inst.toCharArray();

```

```

for(int i = 0; i < array.length; i++)
{
    if(array[i] == 'R')
        x++;
    else if(array[i] == 'L')
        x--;
    else if(array[i] == 'U')
        y++;
    else if(array[i] == 'D')
        y--;
}
return (x == 0 && y == 0);
}
}

```

//Q4. Given two binary strings (strings containing only 1s and 0s) return their sum (also as a binary string).

```

import java.util.*;

public class Q4
{
    static String addBinary(String a, String b)
    {
        if(a.charAt(0) == '0' && b.charAt(0) == '0')
        {
            return "0";
        }

        StringBuilder result = new StringBuilder("");

        int s = 0;

        int i = a.length() - 1, j = b.length() - 1;

        while (i >= 0 || j >= 0 || s == 1)
        {
            s += ((i >= 0)? a.charAt(i) - '0': 0);
            s += ((j >= 0)? b.charAt(j) - '0': 0);

            result.append((char)(s % 2 + '0'));

            s /= 2;

            i--; j--;
        }

        int start = result.length()-1;

        while(start >=0 && result.charAt(start) == '0')
        {
            start--;
        }

        if(start != result.length()-1)

```

```

    {
        result.delete(start+1,result.length());
    }

    return result.reverse().toString();
}

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter first binary string");
    String a = sc.next();
    System.out.println("Enter second binary string");
    String b = sc.next();

    System.out.print(addBinary(a, b));
}
}

```

 //Q5. Note: neither binary string will contain leading 0s unless the string itself is 0

Ex: Given the following binary strings...

"100" + "1", return "101"

"11" + "1", return "100"

"1" + "0", return "1"

```
import java.util.*;
```

```
class Q5
```

```

{
    static String addBinary(String a, String b)
    {

        StringBuilder result = new StringBuilder();

        int aLength = a.length() - 1;
        int bLength = b.length() - 1;

        int carry = 0;

        while(aLength >= 0 || bLength >= 0){

            int sum = carry;

            if(aLength >= 0) {
                sum += (a.charAt(aLength--) - '0');
            }

            if(bLength >= 0) {
                sum += (b.charAt(bLength--) - '0');
            }

            result.insert(0, sum%2);

            carry = (sum / 2);
        }
    }
}

```

```

        if (carry > 0) {
            result.insert(0, 1);
        }

        return result.toString();
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter first binary string");
        String a = sc.next();
        System.out.println("Enter second binary string");
        String b = sc.next();

        System.out.print(addBinary(a, b));
    }
}

```

//Q6. Given an array of strings, return the longest common prefix that is shared amongst all strings.

```

import java.util.*;

class Q6
{
    static String commonPrefixUtil(String str1, String str2)
    {
        String result = "";
        int n1 = str1.length(), n2 = str2.length();

        for (int i = 0, j = 0; i <= n1 - 1 && j <= n2 - 1; i++, j++)
        {
            if (str1.charAt(i) != str2.charAt(j))
            {
                break;
            }
            result += str1.charAt(i);
        }
        return (result);
    }

    static String commonPrefix(String arr[], int n)
    {
        String prefix = arr[0];

        for (int i = 1; i <= n - 1; i++) {
            prefix = commonPrefixUtil(prefix, arr[i]);
        }

        return (prefix);
    }

    public static void main(String[] args)
    {

```

```

Scanner sc = new Scanner(System.in);
System.out.println("Enter input");

String str = sc.nextLine();

String [] arr = null;

arr = str.split(",");

int n = arr.length;

String ans = commonPrefix(arr, n);

if (ans.length() > 0) {
    System.out.printf("The longest common prefix is - %s",
        ans);
} else {
    System.out.printf("There is no common prefix");
}
}
}
}

```

//Q7. Note: you may assume all strings only contain lowercase alphabetical characters.

Ex: Given the following arrays...

["colorado", "color", "cold"], return "col"

["a", "b", "c"], return ""

["spot", "spotty", "spotted"], return "spot"

```

import java.util.*;

class Q7
{
    static String commonPrefixUtil(String str1, String str2)
    {
        String result = "";
        int n1 = str1.length(), n2 = str2.length();

        for (int i = 0, j = 0; i <= n1 - 1 && j <= n2 - 1; i++, j++)
        {
            if (str1.charAt(i) != str2.charAt(j))
            {
                break;
            }
            result += str1.charAt(i);
        }
        return (result);
    }

    static String commonPrefix(String arr[], int n)
    {
        String prefix = arr[0];

        for (int i = 1; i <= n - 1; i++) {
            prefix = commonPrefixUtil(prefix, arr[i]);
        }
    }
}

```

```

        return (prefix);
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter input");

        String str = sc.nextLine().toLowerCase();

        String [] arr = null;

        arr = str.split(",");

        int n = arr.length;

        String ans = commonPrefix(arr, n);

        if (ans.length() > 0) {
            System.out.printf("The longest common prefix is - %s",
                               ans);
        } else {
            System.out.printf("There is no common prefix");
        }
    }
}

```

//Q8. Given a string and the ability to delete at most one character, return whether or not it can form a palindrome.

```

import java.util.*;

class Q8
{
    static boolean isPalindrome(String str, int low, int high)
    {
        while (low < high)
        {
            if (str.charAt(low) != str.charAt(high))
                return false;
            low++;
            high--;
        }
        return true;
    }

    static int possiblePalinAfterRemovingOneChar(String str)
    {
        int low = 0, high = str.length() - 1;

        while (low < high)
        {
            if (str.charAt(low) == str.charAt(high))
            {

```

```

        low++;
        high--;
    }
    else
    {

        if (isPalindrome(str, low + 1, high))
            return low;

        if (isPalindrome(str, low, high - 1))
            return high;
        return -1;
    }
}

return -2;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter any string without space ");
    String str = sc.next();

    int idx = possiblePalinAfterRemovingOneChar(str);

    if (idx == -1)
        System.out.println("It is not possible to convert this String into a Palindrome ");
    else if (idx == -2)
        System.out.println("String is already a Palindrome");
    else
        System.out.println("String can be converted into a Palindrome by removing character at index " + idx);
}
}

```

//Q9. Note: a palindrome is a sequence of characters that reads the same forwards and backwards.

Ex: Given the following strings...

"abcba", return true

"foobof", return true (remove the first 'o', the second 'o', or 'b')

"abccab", return false

```
import java.util.*;
```

```

class Q9
{
    static boolean isPalindrome(String str, int low, int high)
    {
        while (low < high)
        {
            if (str.charAt(low) != str.charAt(high))
                return false;
            low++;
            high--;
        }
        return true;
    }
}

```



```

}

static int possiblePalinAfterRemovingOneChar(String str)
{
    int low = 0, high = str.length() - 1;

    while (low < high)
    {

        if (str.charAt(low) == str.charAt(high))
        {
            low++;
            high--;
        }
        else
        {

            if (isPalindrome(str, low + 1, high))
                return low;

            if (isPalindrome(str, low, high - 1))
                return high;
            return -1;
        }
    }

    return -2;
}

```

```

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter any string without space ");
    String str = sc.next();

    int idx = possiblePalinAfterRemovingOneChar(str);

    if (idx == -1)
        System.out.println("It is not possible to convert this String into a Palindrome ");
    else if (idx == -2)
        System.out.println("String is already a Palindrome");
    else
        System.out.println("String can be converted into a Palindrome by removing character at index " + idx);
}
}

```

//Q10. Given a string representing your stones and another string representing a list of jewels, return the number of stones that you have that are also jewels.

Ex: Given the following jewels and stones...

jewels = "abc", stones = "ac", return 2

jewels = "Af", stones = "AaaddfFf", return 3

jewels = "AYOPD", stones = "ayopd", return 0

```
import java.util.*;
```

```

class Q10
{
    public static int numJewelsInStones(String jewels, String stones)
    {
        int count = 0;
        for(int i=0; i<stones.length();i++)
        {
            for(int j=0;j<jewels.length();j++)
            {
                if(stones.charAt(i) == jewels.charAt(j))
                    count++;
            }
        }
        return count;
    }
}

public static void main (String[] args) throws java.lang.Exception
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter first string");
    String jewels = sc.next();
    System.out.println("Enter second string");
    String stones = sc.next();
    System.out.println(numJewelsInStones(jewels, stones));
}
}

```

//Q11. Given two strings, s and t, merge the two strings together alternating characters starting with s.
 Note: If one string is longer than the other, append the remaining characters of the longer string at the end of the merged string.
 s = "abc", t = "def", return "adbecf".

```

import java.util.*;

class Q11
{
    static String mergeAlternately(String word1, String word2)
    {
        final int n = Math.min(word1.length(), word2.length());
        StringBuilder sb = new StringBuilder();

        for (int i = 0; i < n; ++i)
        {
            sb.append(word1.charAt(i));
            sb.append(word2.charAt(i));
        }
        return sb.append(word1.substring(n)).append(word2.substring(n)).toString();
    }

    public static void main (String [] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter first string");
        String str1 = sc.next();
        System.out.println("Enter second string");
    }
}

```

```
String str2 = sc.next();
System.out.println(mergeAlternately(str1, str2));
}
}
```

//Q12. Given a string, s, return the total number of substring within s that contain the same character.

Note: You may assume that s only contains lowercase alphabetical characters.

Ex: Given the following string s...

s = "aabca", return 6 ("a" appears 3 times, "aa" appears 1 time, "b" appears 1 time, and "c" appears 1 time).

```
import java.util.*;
```

```
class Q12
```

```
{
    static final int MAX_CHAR = 256;
```

```
    static void repeatingChar(String str)
    {
        int count[] = new int[MAX_CHAR];
```

```
        int len = str.length();
```

```
        for (int i = 0; i < len; i++)
        {
            count[str.charAt(i)]++;
        }
```

```
        char ch[] = new char[str.length()];
```

```
        for (int i = 0; i < len; i++)
        {
            ch[i] = str.charAt(i);
            int find = 0;
            for (int j = 0; j <= i; j++)
            {
                if (str.charAt(i) == ch[j])
                    find++;
            }
```

```
            if (find == 1)
                System.out.println(str.charAt(i)+ " appears " + count[str.charAt(i)]+ " times ");
        }
    }
```

```
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string");
```

```
        String str = sc.next();
```

```
        repeatingChar(str);
    }
}
```

//13. Given an encoded string, s, return its decoded representation. The string s will be encoded as follows N[letters], meaning that the letters should be repeated N times in the decoded representation.

Note: You may assume s always encoded correctly (i.e. correct formatting of brackets, only positive values outside the brackets, and always lowercase alphabetical characters inside the brackets).

Ex: Given the following string s...

s = "3[a]2[b]1[c]", return "aaabbc" ("a" is repeated 3 times, "b" is repeated 2 times, and "c" is repeated 1 time).

```
import java.util.*;
```

```
public class Q13
```

```
{
    static String decodeString(String str)
    {
        Stack<Integer> integerstack = new Stack<>();
        Stack<Character> stringstack = new Stack<>();
        String temp = "", result = "";

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

            if (Character.isDigit(str.charAt(i)))
            {
                while (Character.isDigit(str.charAt(i)))
                {
                    count = count * 10 + str.charAt(i) - '0';
                    i++;
                }

                i--;
                integerstack.push(count);
            }

            else if (str.charAt(i) == ']')
            {
                temp = "";
                count = 0;

                if (!integerstack.isEmpty())
                {
                    count = integerstack.peek();
                    integerstack.pop();
                }

                while (!stringstack.isEmpty() && stringstack.peek() != '[' )
                {
                    temp = stringstack.peek() + temp;
                    stringstack.pop();
                }

                if (!stringstack.empty() && stringstack.peek() == '[')
                    stringstack.pop();

                for (int j = 0; j < count; j++)
```

```

        result = result + temp;

        for (int j = 0; j < result.length(); j++)
            stringstack.push(result.charAt(j));

        result = "";
    }

    else if (str.charAt(i) == '[')
    {
        if (Character.isDigit(str.charAt(i-1)))
            stringstack.push(str.charAt(i));

        else
        {
            stringstack.push(str.charAt(i));
            integerstack.push(1);
        }
    }

    else
        stringstack.push(str.charAt(i));
    }

    while (!stringstack.isEmpty())
    {
        result = stringstack.peek() + result;
        stringstack.pop();
    }

    return result;
}

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a string in given format N[char]");

    String str = sc.next();
    System.out.println(decodeString(str));
}
}

```

//Q14. You are given a list of strings, times, where each string represent a timestamp of a twenty-four hour clock (i.e . hours and minutes - "HH:MM"). Return the minimum difference, in minutes, between any two of the timestamps in the list.

Ex: Given the following times...

times = ["01:00", "01:10"], return 10 (i.e. there are 10 minutes between the two times).

Ex: Given the following times...

times = ["00:00", "12:23", "05:50", "23:12"], return 48.

```
import java.util.*;
```

```
public class Q14
{
```

```

public int findMinDifference(List<String> timePoints)
{
    System.out.println("times = " + Arrays.toString(timePoints.toArray()));
    int minimum = Integer.MAX_VALUE;

    List<Integer> minutesList = getMins(timePoints);

    Collections.sort(minutesList);

    int first = minutesList.get(0);
    int last = minutesList.get(minutesList.size()-1);

    minimum = Math.min(minimum, first+1440-last);
    int prev = minutesList.get(0);

    for(int i=1;i<minutesList.size();i++)
    {
        int x = minutesList.get(i)-prev;
        int y = prev + 1440 - minutesList.get(i);
        minimum = Math.min(Math.min(x, y), minimum);
        prev = minutesList.get(i);
    }

    System.out.println("minimum time difference: " + minimum);
    return minimum;
}

private List<Integer> getMins(List<String> timePoints)
{
    List<Integer> minutesList = new ArrayList<>();
    for(int i=0;i<timePoints.size();i++)
    {
        String[] temp = timePoints.get(i).split(":");
        int hour = Integer.parseInt(temp[0]);
        int min = Integer.parseInt(temp[1]) + 60 * hour;
        minutesList.add(min);
    }
    return minutesList;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);

    String str = sc.nextLine();

    String filtered = str.replaceAll("\\[[\\]]\\|\"", "");

    List<String> timePoints = new ArrayList<String>(Arrays.asList(filtered.split(",")));

    TimeDiff m = new TimeDiff();

    m.findMinDifference(timePoints);
}

```

```
}
```

//Q15. Given a string, s, return the length of the longest substring that contains every vowel occurring an even number of times.

Note: You may assume s only contains lowercase alphabetical characters and the vowels you must account for are a, e, i, o, and u.

Ex: Given the following string s...

s = "aeiouaeioua", return 10 (the last 'a' cannot count).

Ex: Given the following string s...

s = "bbb", return 3 (all vowels occur an even number of times, i.e. zero times each).

```
import java.util.*;
```

```
class Q15
```

```
{
```

```
    public static int findLongestSubstring(String s)
```

```
    {
```

```
        int[] map = new int[32];
```

```
        Arrays.fill(map, -1);
```

```
        int state = 0;
```

```
        map[state] = 0;
```

```
        int maxLength = 0;
```

```
        int length = s.length();
```

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

```
        {
```

```
            char c = s.charAt(i);
```

```
            if (c == 'a')
```

```
                state ^= (1 << 0);
```

```
            else if (c == 'e')
```

```
                state ^= (1 << 1);
```

```
            else if (c == 'i')
```

```
                state ^= (1 << 2);
```

```
            else if (c == 'o')
```

```
                state ^= (1 << 3);
```

```
            else if (c == 'u')
```

```
                state ^= (1 << 4);
```

```
            if (map[state] >= 0)
```

```
                maxLength = Math.max(maxLength, i + 1 - map[state]);
```

```
            else
```

```
                map[state] = i + 1;
```

```
        }
```

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

```
        return maxLength;
```

```
    }
```

```
public static void main(String[] args)
```

```
{
```

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

```
    System.out.println("Enter a string");
```

```
    String str = sc.next();
```

```
    findLongestSubstring(str);
```

```
}
```

```
}
```

//Q16. You are given a list of words and asked to find the longest chain. Two words (or more) form a chain if a single letter can be added anywhere in a word, s, to form a word, t (where s and t are both words within the list of words you're given). Return the length of the longest chain you can form.

Ex: Given the following words...

words = ["a", "ab", "abc"], return 3 ("a" can be transformed to "ab" by adding a "b" and "ab" can be transformed by adding a "c" giving a chain length of 3).

Ex: Given the following words...

words = ["a", "abc"], return 1 (both "a" or "abc" form their own chains of size one, they cannot be added together).

```
import java.util.*;
```

```
public class Q16
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

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

```
        String str = sc.nextLine();
```

```
        String filtered = str.replaceAll("\\[[\\]]|\\'| \"\", \"\");
```

```
        String data [] = filtered.split(",");
```

```
        int noOfChanges = 0;
```

```
        int max = 0;
```

```
        int maxIndex = 0;
```

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

```
            if(max < data[i].length()){
```

```
                max = data[i].length();
```

```
                maxIndex = i;
```

```
            }
```

```
        }
```

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

```
            if(maxIndex != i){
```

```
                String temp = data[i];
```

```
                noOfChanges += ( max - temp.length());
```

```
            }
```

```
        }
```

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

```
    }
```

```
}
```

//Q17. 17. You are given two string arrays, queries and words. For any string, s, f(s) is equal to the number of times the smallest lexicographical character occurs in s. For each query, queries[i] count the total number of words where f(queries[i]) < f(word) and return the answer as an array.

Note: Both queries and words will only contain lowercase alphabetical characters and contain at most ten strings each.

Ex: Given the following queries and words...

queries = ["abc"], words = ["def"], return 0 ('a' and 'd' both occur once so f("abc") and f("def") are equal).

Ex: Given the following queries and words...

queries = ["abc"], words = ["ddef", "xyz"], return 2 ('a' appears once and 'd' and 'x' appear twice so f("abc") is less than f("ddef") and f("xyz").)


```
han both f("ddef") and f("xyz")).
import java.util.*;
```

```
public class Q17
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter queries");
        String str1 = sc.nextLine();

        String filtered1 = str1.replaceAll("\\[[\\]]|\\\"|'", "");

        String queries [] = filtered1.split(",");

        System.out.println("Enter words");
        String str2 = sc.nextLine();

        String filtered2 = str2.replaceAll("\\[[\\]]|\\\"|'", "");

        String words [] = filtered2.split(",");

        HashMap<Character,Integer> queriesMap=new HashMap<Character,Integer>();

        HashMap<Character,Integer> wordsMap=new HashMap<Character,Integer>();

        int totalWords = 0;
        int totalQueries = 0;
        for(int i = 0;i<queries.length;i++){
            String temp = queries[i];
            for(int j = 0; j<temp.length();j++){
                char ch = temp.charAt(j);
                if(queriesMap.containsKey(ch)){
                    int total = queriesMap.get(ch);
                    queriesMap.put(ch,++total);
                    if(totalQueries<total){
                        totalQueries = total;
                    }
                }else{
                    queriesMap.put(ch,1);
                }
            }
        }

        for(int i = 0;i<words.length;i++){
            String temp = words[i];
            for(int j = 0; j<temp.length();j++){
                char ch = temp.charAt(j);
                if(wordsMap.containsKey(ch)){
                    int total = wordsMap.get(ch);
                    wordsMap.put(ch,++total);
                    if(totalWords<total){
                        totalWords = total;
                    }
                }else{
```

```

        wordsMap.put(ch,1);
    }
}

if(totalWords==totalQueries){
    System.out.println(0);
}else if (totalQueries<totalWords){
    System.out.println(totalWords);
}else{
    System.out.println(totalQueries);
}
}
}
}

```

Collection Programs

//1. Write a Java program to create a new array list, add some colors (string) and print out the collection.

```

import java.util.*;
public class collection1 {
    public static void main(String[] args) {
        List<String> list_Strings = new ArrayList<String>();
        list_Strings.add("Red");
        list_Strings.add("Green");
        list_Strings.add("Orange");
        list_Strings.add("White");
        list_Strings.add("Black");
        System.out.println(list_Strings);
    }
}

```

//2. Write a Java program to insert an element into the array list at the first position.

```

import java.util.*;
public class collection2 {
    public static void main(String[] args) {
        // Creae a list and add some colors to the list
        List<String> list_Strings = new ArrayList<String>();
        list_Strings.add("Red");
        list_Strings.add("Green");
        list_Strings.add("Orange");
        list_Strings.add("White");
        list_Strings.add("Black");
        // Print the list
        System.out.println(list_Strings);
        // Now insert a color at the first and last position of the list
        list_Strings.add(0, "Pink");
        list_Strings.add(5, "Yellow");
        // Print the list
        System.out.println(list_Strings);
    }
}

```

```
}
```

//3. Write a Java program to retrieve an element (at a specified index) from a given array list.

```
import java.util.*;
public class Collection3 {
    public static void main(String[] args) {
        // Creae a list and add some colors to the list
        List<String> list_Strings = new ArrayList<String>();
        list_Strings.add("Red");
        list_Strings.add("Green");
        list_Strings.add("Orange");
        list_Strings.add("White");
        list_Strings.add("Black");
        // Print the list
        System.out.println(list_Strings);
        // Retrive the first and third element
        String element = list_Strings.get(0);
        System.out.println("First element: "+element);
        element = list_Strings.get(2);
        System.out.println("Third element: "+element);
    }
}
```

//4. Write a Java program to sort a given array list.

```
import java.util.*;
public class Collection4 {
    public static void main(String[] args) {
        // Creae a list and add some colors to the list
        List<String> list_Strings = new ArrayList<String>();
        list_Strings.add("Red");
        list_Strings.add("Green");
        list_Strings.add("Orange");
        list_Strings.add("White");
        list_Strings.add("Black");
        System.out.println("List before sort: "+list_Strings);
        Collections.sort(list_Strings);
        System.out.println("List after sort: "+list_Strings);
    }
}
```

//5. Write a Java program to reverse elements in a array list.

```
import java.util.*;
public class collection5 {
    public static void main(String[] args) {
        // Creae a list and add some colors to the list
        List<String> list_Strings = new ArrayList<String>();
        list_Strings.add("Red");
        list_Strings.add("Green");
        list_Strings.add("Orange");
        list_Strings.add("White");
        list_Strings.add("Black");
        System.out.println("List before reversing :\n" + list_Strings);
    }
}
```

```

Collections.reverse(list_Strings);
System.out.println("List after reversing :\n" + list_Strings);
}
}

```

//6. Write a Java program of swap two elements in an array list.

```

import java.util.ArrayList;
import java.util.Collections;
public class Coll6 {
    public static void main(String[] args) {
        ArrayList<String> c1= new ArrayList<String>();
        c1.add("Red");
        c1.add("Green");
        c1.add("Black");
        c1.add("White");
        c1.add("Pink");

        System.out.println("Array list before Swap:");
        for(String a: c1){
            System.out.println(a);
        }
        //Swapping 1st(index 0) element with the 3rd(index 2) element
        Collections.swap(c1, 0, 2);
        System.out.println("Array list after swap:");
        for(String b: c1){
            System.out.println(b);
        }
    }
}

```

//7. Write a Java program to print all the elements of a ArrayList using the position of the elements.

```

import java.util.ArrayList;
public class Coll7 {
    public static void main(String[] args) {
        ArrayList <String> c1 = new ArrayList <String> ();
        c1.add("Red");
        c1.add("Green");
        c1.add("Black");
        c1.add("White");
        c1.add("Pink");
        System.out.println("\nOriginal array list: " + c1);
        System.out.println("\nPrint using index of an element: ");
        int no_of_elements = c1.size();
        for (int index = 0; index < no_of_elements; index++)
            System.out.println(c1.get(index));
    }
}

```

//8. Write a Java program to append the specified element to the end of a linked list.

```

import java.util.LinkedList;
public class Coll8 {

```

```

public static void main(String[] args) {
    // create an empty linked list
    LinkedList<String> l_list = new LinkedList<String>();
    // use add() method to add values in the linked list
    l_list.add("Red");
    l_list.add("Green");
    l_list.add("Black");
    l_list.add("White");
    l_list.add("Pink");
    l_list.add("Yellow");

    // print the list
    System.out.println("The linked list: " + l_list);
}
}

```

 //9. Write a Java program to insert the specified element at the specified position in the linked list.

```

import java.util.LinkedList;
public class Coll9 {
    public static void main(String[] args) {
        // create an empty linked list
        LinkedList <String> l_list = new LinkedList <String> ();
        // use add() method to add values in the linked list
        l_list.add("Red");
        l_list.add("Green");
        l_list.add("Black");
        l_list.add("White");
        l_list.add("Pink");
        System.out.println("Original linked list: ");
        System.out.println("Let add the Yellow color after the Red Color: " + l_list);
        l_list.add(1, "Yellow");
        // print the list
        System.out.println("The linked list:" + l_list);
    }
}

```

 //10. Write a Java program to insert elements into the linked list at the first and last position.

```

import java.util.LinkedList;
public class Coll10 {
    public static void main(String[] args) {
        // create an empty linked list
        LinkedList<String> l_list = new LinkedList<String>();
        // use add() method to add values in the linked list
        l_list.add("Red");
        l_list.add("Green");
        l_list.add("Black");
        System.out.println("Original linked list:" + l_list);
        // Add an element at the beginning
        l_list.addFirst("White");

        // Add an element at the end of list
    }
}

```

```

l_list.addLast("Pink");
System.out.println("Final linked list:" + l_list);
}
}

```

//11. Write a Java program to display the elements and their positions in a linked list.

```

import java.util.LinkedList;
import java.util.Iterator;
public class Coll11 {
    public static void main(String[] args) {
        // create an empty linked list
        LinkedList<String> l_list = new LinkedList<String>();
        // use add() method to add values in the linked list
        l_list.add("Red");
        l_list.add("Green");
        l_list.add("Black");
        l_list.add("Pink");
        l_list.add("orange");

        // print original list
        System.out.println("Original linked list:" + l_list);
        for(int p=0; p < l_list.size(); p++)
        {
            System.out.println("Element at index "+p+": "+l_list.get(p));
        }
    }
}

```

//12. Write a Java program to check if a particular element exists in a linked list.

```

import java.util.*;
public class Coll12 {
    public static void main(String[] args) {
        // create an empty linked list
        LinkedList <String> c1 = new LinkedList <String> ();
        c1.add("Red");
        c1.add("Green");
        c1.add("Black");
        c1.add("White");
        c1.add("Pink");
        System.out.println("Original linked list: " + c1);

        // Checks whether the color "Green" exists or not.
        if (c1.contains("Green")) {
            System.out.println("Color Green is present in the linked list.");
        } else {
            System.out.println("Color Green is not present in the linked list.");
        }

        // Checks whether the color "Orange" exists or not.
        if (c1.contains("Orange")) {
            System.out.println("Color Orange is present in the linked list.");
        } else {
            System.out.println("Color Orange is not present in the linked list.");
        }
    }
}

```

```
}  
  
}  
}
```

//13. Write a Java program to compare two linked lists.

```
import java.util.*;  
public class Coll13 {  
    public static void main(String[] args) {  
        LinkedList<String> c1= new LinkedList<String>();  
        c1.add("Red");  
        c1.add("Green");  
        c1.add("Black");  
        c1.add("White");  
        c1.add("Pink");  
  
        LinkedList<String> c2= new LinkedList<String>();  
        c2.add("Red");  
        c2.add("Green");  
        c2.add("Black");  
        c2.add("Orange");  
  
        //comparison output in linked list  
        LinkedList<String> c3 = new LinkedList<String>();  
        for (String e : c1)  
            c3.add(c2.contains(e) ? "Yes" : "No");  
        System.out.println(c3);  
    }  
}
```

//14. Write a Java program to replace an element in a linked list.

```
import java.util.LinkedList;  
import java.util.Collections;  
public class Coll14 {  
    public static void main(String[] args) {  
        LinkedList<String> c1= new LinkedList<String>();  
        c1.add("Red");  
        c1.add("Green");  
        c1.add("Black");  
        c1.add("White");  
        c1.add("Pink");  
        System.out.println("Original linked list: " + c1);  
        // Replacing 2nd element with new value  
        c1.set(1, "Orange");  
        System.out.println("The value of second element changed.");  
        System.out.println("New linked list: " + c1);  
    }  
}
```

//15. Write a Java program to iterate through all elements in a hash list.

```
import java.util.*;  
import java.util.Iterator;
```

```

public class Coll15 {
public static void main(String[] args) {
    // Create a empty hash set
    HashSet<String> h_set = new HashSet<String>();
    // use add() method to add values in the hash set
    h_set.add("Red");
    h_set.add("Green");
    h_set.add("Black");
    h_set.add("White");
    h_set.add("Pink");
    h_set.add("Yellow");

    // set Iterator
    Iterator<String> p = h_set.iterator();
    // Iterate the hash set
    while (p.hasNext()) {
        System.out.println(p.next());
    }
}
}

```

//16. Write a Java program to empty an hash set.

```

import java.util.*;

public class Coll16 {
public static void main(String[] args) {
    // Create a empty hash set
    HashSet<String> h_set = new HashSet<String>();
    // use add() method to add values in the hash set
    h_set.add("Red");
    h_set.add("Green");
    h_set.add("Black");
    h_set.add("White");
    h_set.add("Pink");
    h_set.add("Yellow");
    System.out.println("Original Hash Set: " + h_set);
    // Remove all elements
    h_set.removeAll(h_set);
    System.out.println("Hash Set after removing all the elements "+h_set);
}
}

```

//17. Write a Java program to convert a hash set to an array.

```

import java.util.*;

public class Coll17 {
public static void main(String[] args) {
    // Create a empty hash set
    HashSet<String> h_set = new HashSet<String>();
    // use add() method to add values in the hash set
    h_set.add("Red");
    h_set.add("Green");
    h_set.add("Black");
    h_set.add("White");
}
}

```



```

        h_set.add("Pink");
        h_set.add("Yellow");
        System.out.println("Original Hash Set: " + h_set);
        // Creating an Array
        String[] new_array = new String[h_set.size()];
        h_set.toArray(new_array);

        // Displaying Array elements
        System.out.println("Array elements: ");
        for(String element : new_array){
            System.out.println(element);
        }
    }
}

```

//18. Write a Java program to compare two sets and retain elements which are same on both sets.

```

import java.util.*;
public class Coll18 {
    public static void main(String[] args) {
        // Create a empty hash set
        HashSet<String> h_set1 = new HashSet<String>();
        // use add() method to add values in the hash set
        h_set1.add("Red");
        h_set1.add("Green");
        h_set1.add("Black");
        h_set1.add("White");
        System.out.println("Frist HashSet content: "+h_set1);
        HashSet<String>h_set2 = new HashSet<String>();
        h_set2.add("Red");
        h_set2.add("Pink");
        h_set2.add("Black");
        h_set2.add("Orange");
        System.out.println("Second HashSet content: "+h_set2);
        h_set1.retainAll(h_set2);
        System.out.println("HashSet content:");
        System.out.println(h_set1);
    }
}

```

//19. Write a Java program to create a new tree set, add some colors (string) and print out the tree set.

```

import java.util.TreeSet;
public class Coll19 {
    public static void main(String[] args) {
        TreeSet<String> tree_set = new TreeSet<String>();
        tree_set.add("Red");
        tree_set.add("Green");
        tree_set.add("Orange");
        tree_set.add("White");
        tree_set.add("Black");
        System.out.println("Tree set: ");
        System.out.println(tree_set);
    }
}

```

//20. Write a Java program to find the numbers less than 7 in a tree set.

```
import java.util.TreeSet;
import java.util.Iterator;

public class Coll20 {
    public static void main(String[] args) {
        // creating TreeSet
        TreeSet<Integer>tree_num = new TreeSet<Integer>();
        TreeSet<Integer>treeheadset = new TreeSet<Integer>();

        // Add numbers in the tree
        tree_num.add(1);
        tree_num.add(2);
        tree_num.add(3);
        tree_num.add(5);
        tree_num.add(6);
        tree_num.add(7);
        tree_num.add(8);
        tree_num.add(9);
        tree_num.add(10);

        // Find numbers less than 7
        treeheadset = (TreeSet)tree_num.headSet(7);

        // create an iterator
        Iterator iterator;
        iterator = treeheadset.iterator();

        //Displaying the tree set data
        System.out.println("Tree set data: ");
        while (iterator.hasNext()){
            System.out.println(iterator.next() + " ");
        }
    }
}
```

//21. Write a Java program to remove all the elements from a priority queue.

```
import java.util.*;
public class Coll21 {
    public static void main(String[] args) {
        // Create Priority Queue
        PriorityQueue<String> pq1 = new PriorityQueue<String>();
        // use add() method to add values in the Priority Queue
        pq1.add("Red");
        pq1.add("Green");
        pq1.add("Black");
        pq1.add("White");
        System.out.println("Original Priority Queue: "+pq1);

        // Removing all the elements from the Priority Queue
        pq1.clear();
    }
}
```

```

        System.out.println("The New Priority Queue: " + pq1);
    }
}

```

//22. Write a Java program to count the number of key-value (size) mappings in a map

```

import java.util.*;
public class Coll22 {
    public static void main(String args[]){
        HashMap<Integer,String> hash_map= new HashMap<Integer,String>();
        hash_map.put(1, "Red");
        hash_map.put(2, "Green");
        hash_map.put(3, "Black");
        hash_map.put(4, "White");
        hash_map.put(5, "Blue");
        System.out.println("Size of the hash map: "+hash_map.size());
    }
}

```

//23. Write a Java program to convert a priority queue to an array containing all of the elements of the queue

```

import java.util.*;

public class Coll23 {
    public static void main(String[] args) {

        // Create Priority Queue
        PriorityQueue<String> pq1 = new PriorityQueue<String>();
        // use add() method to add values in the Priority Queue
        pq1.add("Red");
        pq1.add("Green");
        pq1.add("Black");
        pq1.add("White");
        System.out.println("Original Priority Queue: "+pq1);

        //Convert a linked list to array list
        List<String> array_list = new ArrayList<String>(pq1);
        System.out.println("Array containing all of the elements in the queue: "+array_list);

    }
}

```

//24. Write a Java program to check whether a map contains key-value mappings (empty) or not

```

import java.util.*;
public class Coll24 {
    public static void main(String args[]) {
        HashMap <Integer,String> hash_map = new HashMap <Integer,String> ();
        hash_map.put(1, "Red");
        hash_map.put(2, "Green");
        hash_map.put(3, "Black");
        hash_map.put(4, "White");
        hash_map.put(5, "Blue");
        // check if map is empty
        boolean result = hash_map.isEmpty();
    }
}

```

```
// check the result
System.out.println("Is hash map empty: " + result);
// Removing all the elements from the linked map
hash_map.clear();
// check if map is empty
result = hash_map.isEmpty();
// check the result
System.out.println("Is hash map empty: " + result);
}
}
```

//25. Write a Java program to get the value of a specified key in a map

```
import java.util.*;
public class Coll25 {
    public static void main(String args[]){
        HashMap<Integer,String> hash_map= new HashMap<Integer,String>();
        hash_map.put(1,"Red");
        hash_map.put(2,"Green");
        hash_map.put(3,"Black");
        hash_map.put(4,"White");
        hash_map.put(5,"Blue");
        // get value of key 3
        String val=(String)hash_map.get(3);
        // check the value
        System.out.println("Value for key 3 is: " + val);
    }
}
```
