1. Create a class Savings Account. Use a static variable annualInterestRate to store the annual interest for all accounts holders. Each object of the class contains a private data member savingsBalance indicating the amount of saver currently has on deposit and an account number. Provide a function calculateMonthlyInterest to calculate the monthly interest by multiplying the savings Balance by annualInterest Rate divided by 12, this interest should be added to savings Balance. Provide a static function modifyInterestRate that sets annualInterestRate to a new value. Write a program to test the class Savings Account. Set the annualInterestRate to 4%,then calculate the monthly interst and show the balance of the saver. Then set annualInterestRate to 5% and repeat the above calculation.

```
import java.util.Scanner;

class SavingsAccount
{
    static float annualInterestRate;
    private double savingsBalance;
    private long account_number;
    double i;

    SavingsAccount()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Acc no. :");
        long an = sc.nextLong();
        account_number = an;
        System.out.print("Enter savings balance :");
        double sb = sc.nextDouble();
        savingsBalance = sb;
        System.out.println("\n");
        sc.close();
    }

    public void calculateMonthlyInterest()
    {
        i = ((savingsBalance * annualInterestRate) / (12));
        savingsBalance = savingsBalance + i;
    }

    public static void ModifyInterestRate(float r)
    {
        annualInterestRate = r;
    }

    public void account()
    {
        System.out.println("Acc no. :" + account_number);
        System.out.println("Interest Rate :" + annualInterestRate);
        System.out.println("Interest generated :" + i);
        System.out.println("Balance :" + savingsBalance + "\n");
    }
}

public class demo
{
    public static void main(String[] args)
    {
        SavingsAccount obj = new SavingsAccount();
        SavingsAccount.ModifyInterestRate(4);
        obj.calculateMonthlyInterest();
```

```
        obj.account();
        SavingsAccount.ModifyInterestRate(5);
        obj.calculateMonthlyInterest();
        obj.account();
    }
```

2 WAP to display all the command line arguments.

```
public class cli
{
    public static void main(String[] args)
    {
        for(int i=0; i<args.length; i++)
        {
            System.out.print(args[i] + " ");
        }
    }
}
```

3.WAP to keep the data in a 2-D array value[3][4] and also display the content of array on console.

```java
import java.util.Scanner;

public class two_d_array
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int[][] a = new int[3][4];
        System.out.println("Enter 12 values of matrix ->");
        for(int i=0; i<3; i++)
        {
            for(int j=0; j<4; j++)
            {
                System.out.print("Enter [" + (i+1) + "][" + (j+1) + "] :");
                a[i][j] = sc.nextInt();
            }
        }
        System.out.println("\nMatrix ->");
        for(int i=0; i<3; i++)
        {
            for(int j=0; j<4; j++)
            {
                System.out.print(a[i][j] + "\t");
            }
            System.out.print("\n");
        }
        sc.close();
    }
}
```