

SHERAL SIMON WASKAR

20BCE1182

Lab- Exercise on methods in java

1) Write a JAVA program to find the GCD of any two numbers. Your program should have method findGCD() that return the gcd of the two numbers.

Code :

```
import java.util.Scanner;

class Gcd
{

    public static void main(String arg[])
    {
        long num1,num2;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the first number :");

        num1 = sc.nextLong();

        System.out.println("Enter the second number :");

        num2 = sc.nextLong();

        if(num1>0 && num2>0)
        {
            long res = gcd_cal(num1,num2);
            System.out.println("The GCD of two numbers = " + res);
        }
        else
            System.out.println("Please enter numbers greater than zero");
    }
}
```

```

}

static long gcd_cal(long a ,long b)

{
    while(b>0)
    {
        long t = b;
        b = a % b;
        a = t;
    }
    return a;
}

```

}

OUTPUT:

```

D:\SEM 4\CSE1007_LAB>javac Gcd.java

D:\SEM 4\CSE1007_LAB>java Gcd.java
Enter the first number :
34
Enter the first number :
25
The GCD of two numbers = 1

```

```

D:\SEM 4\CSE1007_LAB>javac Gcd.java

D:\SEM 4\CSE1007_LAB>java Gcd.java
Enter the first number :
25
Enter the first number :
5
The GCD of two numbers = 5

```

2) Travel Tickets Company sells tickets for airlines, tours, and other travel-related services. Because long numbers have often been entered incorrectly by agents, Travel Tickets has asked you to code a JAVA program that will indicate if a ticket number entry is invalid. Ticket numbers are 11 digits long. Ticket numbers are designed such that if you drop the last digit of the number, then divide the 10-digit number by 7, the remainder of the division will be identical to the last dropped digit. If ticket number is 10-digits, include the 11th digit or if it is 11-digit long, check for the validity. If the ticket number is any other length, your program should prompt the agent to check and re-enter the ticket

number. Include a method isValidTicket() that return either true or false depending on the validity of the ticket.

CODE :

```
import java.util.Scanner;

class TicketsValidity
{
    public static int len(long num)
    {
        int count =0;
        while(num!=0)
        {
            num = num / 10;
            count = count+1;
        }
        return count;
    }

    public static boolean isValidTicket(long num)
    {
        long new_num;
        int length = len(num);
        if (length == 10)
        {
            long ldig = num % 7;
            System.out.println("The ticket has 10 digits , last digit is " + ldig);
            num = num*10;
            new_num = num + ldig;
            System.out.println("The valid ticket number is " + new_num);
        }
    }
}
```

```
    }

    else if (length == 11)

    {

        long remdig = num/10;

        long ldig = num % 10;

        boolean s = remdig % 7 == ldig;

        System.out.println("The ticket is " + s);

    }

    return true;
```

```
}

public static void main(String args[])

{

    long num;

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter ticket number :");

    num = sc.nextLong();

    isValidTicket(num);

}

}

OUTPUT :
```

```
D:\SEM 4\CSE1007_LAB>javac TicketsValidity.java

D:\SEM 4\CSE1007_LAB>java TicketsValidity.java
Enter ticket number :
19385728491
The ticket is false

D:\SEM 4\CSE1007_LAB>javac TicketsValidity.java

D:\SEM 4\CSE1007_LAB>java TicketsValidity.java
Enter ticket number :
1938738288
The ticket has 10 digits , last digit is 4
The valid ticket number is 19387382884
```

3) Assume that you have a list of words and you wish to find how many words are palindrome in the list. Devise a JAVA program that reads several words and displays the palindrome words and count of such words. Your program should have two methods namely

String reverseString(String) and boolean isPalindrome(String).

Hint: A word is palindrome if its reverse is same as the original.

CODE :

```
import java.util.Scanner;

class Palindrome
{
    public static void main(String args[])
    {
        String str, rev = "";
        int c=0;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a string:");
        str = sc.nextLine();

        if (isPalindrome(str))
```

```
{  
    c++;  
  
    System.out.println(str+" is a palindrome");  
  
    System.out.println("The count of palindrome string is " +c);  
}  
  
else  
  
    System.out.println(str+" is not a palindrome");  
  
}  
  
  
public static String reverseString(String str)  
{  
  
    String rev="";  
  
    int length = str.length();  
  
  
    for ( int i = length - 1; i >= 0; i-- )  
        rev = rev + str.charAt(i);  
  
    return rev;  
  
}  
  
public static boolean isPalindrome(String str)  
{  
  
    String s;  
  
    s = reverseString( str);  
  
    if (s.equals(str))  
    {  
  
        return true;  
    }  
  
    else  
    {  
  
        return false;  
    }  
}
```

```
    }  
}  
}
```

OUTPUT :

```
D:\SEM 4\CSE1007_LAB>java Palindrome.java  
Enter a string:  
abba  
abba is a palindrome  
The count of palindrome string is 1
```

```
D:\SEM 4\CSE1007_LAB>java Palindrome.java  
Enter a string:  
apple  
apple is not a palindrome
```