**NAME : SHASHIDHAR      MOB.NO.:6361662695     BATCH CODE : BECM71**

## 1. WAJP TO REVERSE THE GIVRN NUMBER

```
public class Revers {

      public static void main(String[] args) {
            int rev=0;
            int a=123;
            while(a!=0) {
                  int rem=a%10;
                  rev=rev*10+rem;
                  a=a/10;
            }
      System.out.println(rev);
      }
}
```

## 2. WAJP TO CHECK THE GIVEN NUMBER IS PALINDROME OR NOT

```
public class Palindrome {

      public static void main(String[] args) {
            int rev=0;
            int a=11211;
            int b=a;
            while(a!=0) {
                  int rem=a%10;
                  rev=rev*10+rem;
                  a=a/10;
            }
        if(rev==b) {
            System.out.println("the entered number is PALINDROME");
        }else {
            System.out.println("the entered number is NOT A PALINDROME");
        }
      }
}
```

## 3. WAJP TO GET COUNT OF DIGITS IN A GIVEN NUMBER

```
public class Count {

      public static void main(String[] args) {
       int a=7806;
       int count=0;
       while(a!=0) {
             count++;
             a=a/10;
       }
     System.out.println("the number of digits is : "+count);
      }
}
```

## 4. WAJP TO COUNT THE NUMBER OF EVEN AND ODD DIGITS IN A GIVEN NUMBER

```
public class CuEvOd {

      public static void main(String[] args) {
            int a=125689;
            int evencount=0;
            int oddcount=0;
            while(a!=0) {
            int rem=a%10;
            if(rem%2==0) {
                  evencount++;
            }else {
                  oddcount++;
            }
            a=a/10;
      }
      System.out.println("EVEN COUNT : "+evencount+" ODD COUNT : "+oddcount);
      }

      }
```

## 5. WAJP TO GET THE SUM OF DIGITS IN A GINEN NUMBER

```
public class Sum {

      public static void main(String[] args) {
            int a=123;
            int sum=0;
            while(a!=0) {
                  int rem=a%10;
                  sum=sum+rem;
                  a=a/10;

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

## 6. WAJP TO PRINT PRIME NUMBERS IN THE GIVEN RANGE

```
public class PrimeRange {

      public static void main(String[] args) {
            int n=100;
            for(int a=1; a<=n; a++) {
                  int count=0;
                  for(int i=1; i<=a; i++) {
                        if(a%i==0) {
                              count++;
                        }
                  }
                  if(count==2) {
                        System.out.println(a);
                  }
            }
      }
      }
```

## 7. WAJP TO PRINT GIVEN NUMBER IS PRIME OR NOT

```
import java.util.Scanner;
public class PrimeYN {

      public static void main(String[] args) {
            Scanner s1=new Scanner(System.in);
            System.out.println("Enter the number");
     int n=s1.nextInt();
     int count=0;
     for(int i=1; i<=n; i++) {
       if(n%i==0) {
             count++;
             s1.close();
       }
     }
     if(count==2) {
       System.out.println("The given number "+n+" is prime number");
     }else {
       System.out.println("The given number "+n+" is not a prime number");
     }
      }

      }
```

## 8. WAJP TO SWAP TWO NUMBERS USING TEMP VARIABLE AND WITHOUT TEMP VERIABLE

```
public class Swap {

      public static void main(String[] args) {
       int a=5;
       int b=10;
       System.out.println("original a value : "+a);
       System.out.println("original b value : "+b);
       int t=a; //a=a+b;     //b=a+b-(a=b);
       a=b;     //b=a-b;
       b=t;    //a=a-b;
       System.out.println("Swaped a value : "+a);
       System.out.println("Swaped b value : "+b);
      }
      }
```

## 9. WAJP TO CHECK THE GIVEN NUMBER IS ARMSTRONG NUMBER OR NOT

```
public class Armstrong {

      public static void main(String[] args) {
            int a=153;
            int b=a;
            int total=0;
            int count=0;
            while(b!=0) {
                  count++;
                  b=b/10;
            }
       int c=a;
       while(c!=0) {
         int eachNumberPower=1;
         int rem=c%10;
         for(int i=1; i<=count; i++) {
               eachNumberPower=eachNumberPower*rem;
         }
         c=c/10;
         total=total+eachNumberPower;
       }
       if(a==total) {
         System.out.println("the given number "+a+" is ARMSTRONG.");
       }
      }

      }
```

## 10.WAJP TO PRINT ARMSTRONG NUMBERS BETWEEN RANGE

```
public class ArmstrongRang {

      public static void main(String[] args) {
            for(int a=1; a<=1000; a++) {
            int b=a;
            int total=0;
            int count=0;
            while(b!=0) {
                  count++;
                  b=b/10;
            }
       int c=a;
       while(c!=0) {
         int eachNumberPower=1;
         int rem=c%10;
         for(int i=1; i<=count; i++) {
               eachNumberPower=eachNumberPower*rem;
         }
         c=c/10;
         total=total+eachNumberPower;
       }
       if(a==total) {
         System.out.println(a);
       }

      }
      }
   }
```

## 11.WAJP TO GET FACTORIAL OF A GIVEN NUMBER

```
public class Fact {

      public static void main(String[] args) {
            int a=4;
            int fact=1;
            for(int i=1; i<=a; i++) {
                  fact=fact*i;
            }
      System.out.println(a+"! is "+fact);
      }

   }
```

## 12.WAJP TO GET FACTORIAL OF A GIVEN RANGE OF NUMBERS

```
public class FactRang {

      public static void main(String[] args) {
            for(int a=1; a<=20; a++) {
            int fact=1;
            for(int i=1; i<=a; i++) {
                  fact=fact*i;
            }
      System.out.println(a+" !   is "+fact);
      }
      }
}
```

## 13.WAJP TO PRINT FIBANOCI SERIES

```
public class Fibanoci {

      public static void main(String[] args) {
            int n1=0;
            int n2=1;
            System.out.print(n1+" "+n2);
            for(int i=1; i<=7; i++) {
       int sum=n1+n2;
       System.out.print(" "+sum);
       n1=n2;
       n2=sum;
      }
      }
   }
```

## 14.WAJP TO GET LARGEST OF 3 NUMBERS

```
public class Lar03 {

      public static void main(String[] args) {
            int a=9, b=5, c=3;
            if(a>b) {
                  if(a>c) {
                        System.out.println("a is larger");
                  }
                  else {
                        System.out.println("c is larger");
                  }
            }
            else if(b>c) {
                  System.out.println("b is larger");
            }
            else {
                  System.out.println("c is larger");
            }
      }
   }
```

## 15.WAJP TO GET LARGEST OF 4 NUMBERS

```java
public class Lrgof4 {

      public static void main(String[] args) {
            int a=70, b=60, c=50, d=40;
            if(a>b) {
                  if(a>c) {
                        if(a>d) {
                              System.out.println("a is largest");
                        }
                        else {
                              System.out.println("d is largest");
                        }
                  }
                  else if(c>d) {
                        System.out.println("c is largest");
                  }
                  else {
                        System.out.println("d is largest");
                  }
            }
            else if(b>c) {
                  if(b>d) {
                        System.out.println("b is largest");
                  }
                  else {
                        System.out.println("d is largest");
                  }
            }
            else if(c>d) {
                  System.out.println("c is largest");
            }
            else {
                  System.out.println("d is largest");
            }
      }

}
```