

```

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.Collection;


/**
 * This HelloJava program takes 1 file as input and
 * it calculates middle reimann sum,arcsin and armstrong numbers.
 * @author Busra Cicek
 *
 */

public class HelloJava{

    public static String[] newInput;


    public static void main(String[] args) {

        //READ FILE

        String[] inputfile = readfile(args[0]);


        //CALL THE METHOD FOR CALCULATION

        HelloJava javaClass=new HelloJava();

        javaClass.newInput=inputfile;

        compare(newInput);

    }

    /**
     * @param path
     * @return arrays as the files
     * @throws IOException
     */

```

```
public static String[] readfile(String path) {
```

```
    //This method reads file as input
```

```
    String[] array = null;
```

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

```
    try {
```

```
        for (String line : Files.readAllLines(Paths.get(path))) {
```

```
            for(String word : line.split(" "))
```

```
                words.add(word);
```

```
        }
```

```
        array = (String[])words.toArray(new String[words.size()]);
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    return array;
```

```
}
```

```
public static void compare(String[] array){
```

```
    int sizeInput=array.length; //sizeInput shows length of array
```

```
    //This method compares elements of newInput array to find result of calculations.
```

```
    int i=0;
```

```
    while(i<=sizeInput){
```

```
        if(array[i].equals("IntegrateReimann")){
```

```
            i+=1;
```

```
            if(array[i].equals("Func1")){
```

```
                //if func1, then first function (f(x)=x^2-x+3) is used for calculation
```

```

String int1=array[i+=1];
String int2=array[i+=1];
String int3=array[i+=1];

double double1=Double.parseDouble(int1);
double double2=Double.parseDouble(int2);
double double3=Double.parseDouble(int3);

System.out.println("IntegrateReimann"+" "+"Func1"+"
"+int1+" "+int2+" "+int3+" "+"Result:"+" "+calculationFunc1(double1,double2,double3));

i+=1;
}
else if(array[i].equals("Func2")){

```

calculation //if func2, then second function ($f(x)=(3\sin(x)-4)^2$) is used for

```

String int1=array[i+=1];
String int2=array[i+=1];
String int3=array[i+=1];

double double1=Double.parseDouble(int1);
double double2=Double.parseDouble(int2);
double double3=Double.parseDouble(int3);

System.out.println("IntegrateReimann"+" "+"Func2"+"
"+int1+" "+int2+" "+int3+" "+"Result:"+" "+calculationFunc2(double1,double2,double3));

i+=1;
}
else if(array[i].equals("Func3")){

```

calculation //if func3, then third function ($f(x)=\operatorname{arcsinh}$) is used for

```

        String int1=array[i+=1];

        String int2=array[i+=1];

        String int3=array[i+=1];


        double double1=Double.parseDouble(int1);

        double double2=Double.parseDouble(int2);

        double double3=Double.parseDouble(int3);

        System.out.println("IntegrateReimann"+" "+"Func3"+"
"+int1+" "+int2+" "+int3+" "+"Result:"+" "+calculationFunc3(double1,double2,double3));

        i+=1;

    }

}

if(array[i].equals("Arcsinh")){

    i+=1;

    String int1=array[i];

    double double1=Double.parseDouble(int1);

    System.out.println("Arcsinh"+" "+int1+" "+"Result:"+"
"+calculationArcsinh(double1));

    i+=1;

}

if(array[i].equals("Armstrong")){

    //This condition writes armstrong numbers.(Armstrong number is is a
number that is the sum of its own digits each raised to the power of the number of digits.)

    i+=1;

    String String1=array[i];

    int int1=Integer.parseInt(String1);

    System.out.print("Armstrong"+" "+String1+" "+"Result: ");

    calculationArmstrong(int1);

    i+=1;

}

}

```

```

        if(i>sizeInput-1){
            break;
        }

    }

}

public static double calculationFunc1(double iLow,double iUp,double iRec){

    // calculates Function1

    double x,power,diff,rfunc,result=0;
    diff=(iUp-iLow)/iRec;
    double xNum=iLow+diff;
    int i=0;
    for(x=(xNum+(xNum-diff))/2; x<=iUp;){
        while(i<iRec){
            power=x*x;
            rfunc=power-x+3;
            result=result+rfunc;
            x=x+diff;
            i+=1;
        }

    }

    return(diff*result);
}

public static double calculationFunc2(double iLow,double iUp,double iRec){

    //calculates Funcion2

```

```

double x,diff,iSin,dNum,rfunc,result=0;

diff=(iUp-iLow)/iRec;

double xNum=iLow+diff;

int i=0;

for(x=(xNum+(xNum-diff))/2; x<=iUp;){

    while(i<iRec){

        iSin=3*(Math.sin(x));

        dNum=iSin-4;

        rfunc=dNum*dNum;

        result=result+rfunc;

        x=x+diff;

        i+=1;

    }

}

return (diff*result);

}

public static double calculationFunc3(double iLow,double iUp,double iRec){

    //calculates Function3

    double x,diff,power,factorial,power1,factorial1,valuePow,rfunc,result=0;

    diff=(iUp-iLow)/iRec;

    for(double i=iUp; i>=iLow; i-=diff){

        for(double n=0; n<=30; n++){

            x=(i+(i-diff))/2;

            power=Math.pow((-1), n);

            factorial=calculationFact(2*n);

            power1=Math.pow(4, n);

            factorial1=calculationFact(n)*calculationFact(n);

            valuePow=Math.pow(x, 2*n+1);

```

```

        rfunc=(power*factorial*valuePow)/(power1*factorial1*(2*n+1));
        result=result+rfunc;
    }
}
return (diff*result);
}

```

```

public static double calculationArcsinh(double value){

```

```

    //calculates Arcsinh

```

```

    double power,factorial,power1,factorial1,valuePow,rfunc,result=0;
    for(double n=0; n<=30; n++){
        power=Math.pow((-1), n);
        factorial=calculationFact(2*n);
        power1=Math.pow(4, n);
        factorial1=calculationFact(n)*calculationFact(n);
        valuePow=Math.pow(value, 2*n+1);
        rfunc=(power*factorial*valuePow)/(power1*factorial1*(2*n+1));
        result=result+rfunc;
    }
    return result;

```

```

}

public static void calculationArmstrong(int value){

```

```

    //calculates Armstrong numbers

```

```

    int number,iResult;
    for(number=0; number>=0; number++){
        iResult=calculationRemainder(number,value);
        if(number==iResult){

```

```

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

}

public static int calculationRemainder(int number,int value){

    //calculates remainder for Armstrong numbers

    int iDigit,remainder,result=0;
    iDigit=calculationDigit(number);
    if (iDigit<=value){
        //return the loop until iDigit>value
        while(number>0){
            remainder=number%10;
            result=(int) ((int)result+Math.pow(remainder, iDigit));
            number=number/10;
        }
    }
    return result;

}

public static int calculationDigit(int number){

    //calculates digit of number for Armstrong numbers
    int digit=0;
    while(number>0){
        //if number>0, then add 1 to digit
        number=number/10;
    }
}

```



```
        digit++;
    }

    return digit;

}

public static double calculationFact(double n){

    //calculates factorial
    double fact=1;
    if(n==0){

        //0!=1

        fact=1;
    }
    else{
        for(double a=1; a<=n; a++){
            fact=fact*(a);
        }
    }
    return fact;

}

}
```