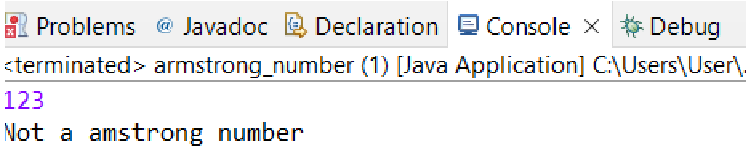
- Here we are using methods to solve all problems

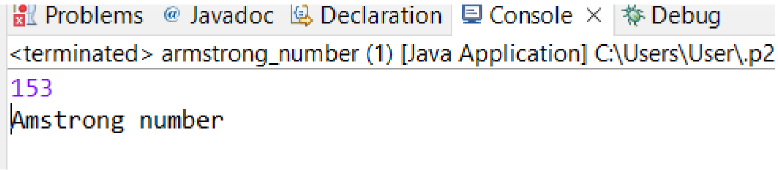
armstrong_number:

```
public class armstrong_number {

static void armstrong() {
	Scanner sc=new Scanner(System.in);
	int num=sc.nextInt();
	int temp=num;
	int temp3=num;
	int count=0;
	while(num!=0) {
		int last=num%10;
		count++;
		num=num/10;
	}
	double d=0;

	while(temp!=0) {
		int temp1=temp%10;
		d+=Math.pow(temp1, count);
		temp=temp/10;
	}

	if(temp3==d) {System.out.println("Amstrong number");}
	else {System.out.println("Not a amstrong number");}
}
public static void main(String[] args) {

	armstrong();
}

}
```

Out put:



```
<terminated> armstrong_number (1) [Java Application] C:\Users\User\.
123
Not a amstrong number
```

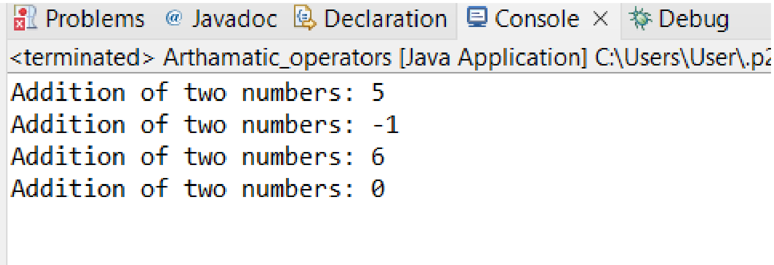


```
<terminated> armstrong_number (1) [Java Application] C:\Users\User\.p2
153
Amstrong number
```

Arthamatic_operators:

```
    public class Arthamatic_operators {

    static void add(int i,int j) {
        int k=i+j;
        System.out.println("Addition of two numbers: "+k);
    }
    static void sub(int i,int j) {
        int k=i-j;
        System.out.println("Addition of two numbers: "+k);
    }
    static void mul(int i,int j) {
        System.out.println("Addition of two numbers: 
"+i*j);
    }
    static void div(int i,int j) {
        System.out.println("Addition of two numbers: 
"+i/j);
    }
    public static void main(String[] args) {

        add(2,3);
        sub(2,3);
        mul(2,3);
        div(2,3);

    }

}
```

Out put:



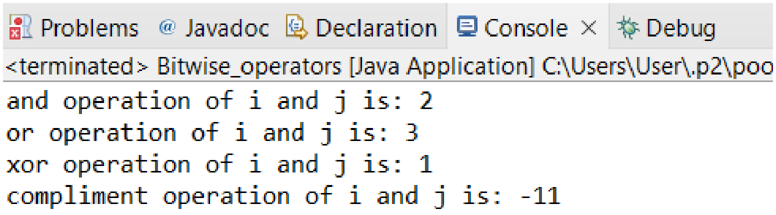
<terminated> Arthamatic_operators [Java Application] C:\Users\User\.p2

```
Addition of two numbers: 5
Addition of two numbers: -1
Addition of two numbers: 6
Addition of two numbers: 0
```

Bitwise_operators:

```
public class Bitwise_operators {

	static void and(int i,int j) {
		int k=i&j;
		System.out.println("and operation of i and j is: "+k);
	}
	static void or(int i,int j) {
		int k=i|j;
		System.out.println("or operation of i and j is: "+k);
	}
	static void xor(int i,int j) {
		int k=i^j;
		System.out.println("xor operation of i and j is: "+k);
	}
	static void compliment(int i) {
		int k=~i;
		System.out.println("compliment operation of i and j is: 
"+k);
	}
	public static void main(String[] args) {
		and(2,3);
		or(2,3);
		xor(2,3);
		compliment(10);

	}

	}
```

Output :

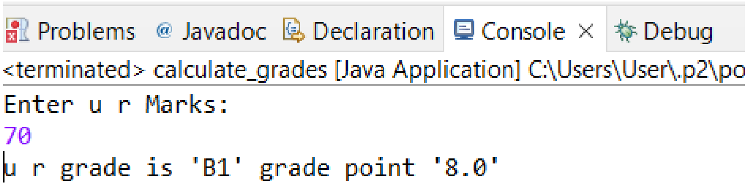


```
<terminated> Bitwise_operators [Java Application] C:\Users\User\.p2\poo
and operation of i and j is: 2
or operation of i and j is: 3
xor operation of i and j is: 1
compliment operation of i and j is: -11
```

calculate_grades:

```
public class calculate_grades {
	static void grades() {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter u r Marks:");
		int d=sc.nextInt();
		d=d/10;
		switch(d) {
		case 1:case 2:case 0: System.out.println("u r grade is 'F' grade point 'O'"); break;
		case 3: System.out.println("u r grade is 'D' grade point '4.0'"); break;
		case 4: System.out.println("u r grade is 'C2' grade point '5.0'");
		case 5: System.out.println("u r grade is 'C1' grade point '6.0'"); break;
		case 6: System.out.println("u r grade is 'B2' grade point '7.0'"); break;
		case 7: System.out.println("u r grade is 'B1' grade point '8.0'"); break;
		case 8: System.out.println("u r grade is 'A2' grade point '9.0'"); break;
		case 10:case 9: System.out.println("u r grade is 'A1' grade point '10.0'"); break;
		default: System.out.println("Please enter marks from 0 to 100");
		}

	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		grades();
	}

}
```

Output:



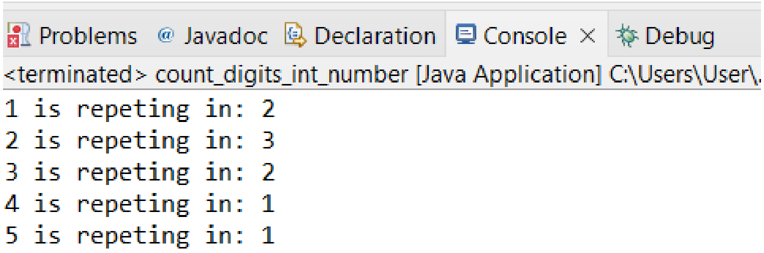
<terminated> calculate_grades [Java Application] C:\Users\User\.p2\po

```
Enter u r Marks:
70
u r grade is 'B1' grade point '8.0'
```

count_digits_int_number:

```
public class count_digits_int_number {

        static void count(int i) {
                int count=0;
                int k=i;
                while(i!=0)
                {
                        count++;
                        i=i/10;
                }

                int a[]=new int[count];
                for(int j=0;j<count;j++) {
                        int h=k%10;
                        a[j]=h;
                        k/=10;
                }
                Arrays.sort(a);
                int temp=1;
                for(int m=0;m<a.length;m++) {
                        for(int n=m+1;n<a.length;n++) {
                                if(a[m]==a[n]) {
                                        temp++;
                                        m++;
                                }
                        }
                        System.out.println(a[m]+" is repeting in: "+temp);
                        temp=1;
                }
        }
        public static void main(String[] args) {
                count(112233245);

        }

}
```

Output:



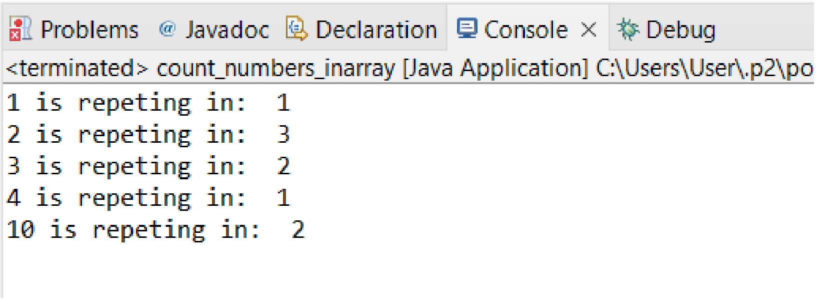
<terminated> count_digits_int_number [Java Application] C:\Users\User\.

```
1 is repeting in: 2
2 is repeting in: 3
3 is repeting in: 2
4 is repeting in: 1
5 is repeting in: 1
```

count_numbers_inarray:

```
public class count_numbers_inarray {

    static void count(int arr[]) {
        Arrays.sort(arr);
        int count=1;
        for(int i=0;i<arr.length;i++) {
            for(int j=i+1;j<arr.length;j++) {

                if(arr[i]==arr[j]) {
                    count++;
                    i++;
                    }
            }
            System.out.println(arr[i]+" is repeting in:
"+count);
            count=1;
         }
    }
    public static void main(String[] args) {

        int arr1[]= {1,2,10,3,4,3,2,10,2};
        count(arr1);

    }

    }
```

Output:



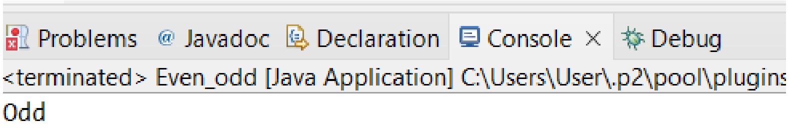
```
<terminated> count_numbers_inarray [Java Application] C:\Users\User\.p2\po
1 is repeting in:  1
2 is repeting in:  3
3 is repeting in:  2
4 is repeting in:  1
10 is repeting in:  2
```

Even_odd:

```
public class Even_odd {

	static void number(int i) {
		switch(i%2) {
		case 0:
			System.out.println("Even");
			break;
		default:
			System.out.println("Odd");
			break;
		}
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		number(11);
	}

}
```

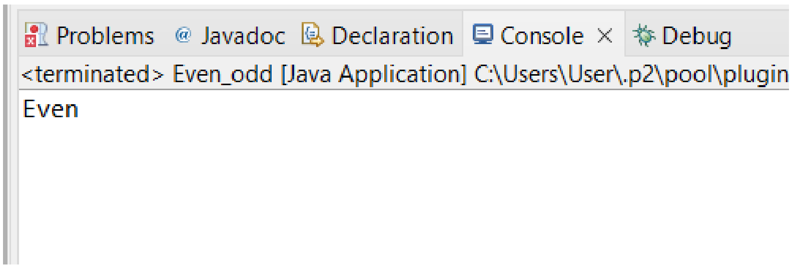
Output:

Input: 11



```
<terminated> Even_odd [Java Application] C:\Users\User\.p2\pool\plugins
Odd
```

Input: 10

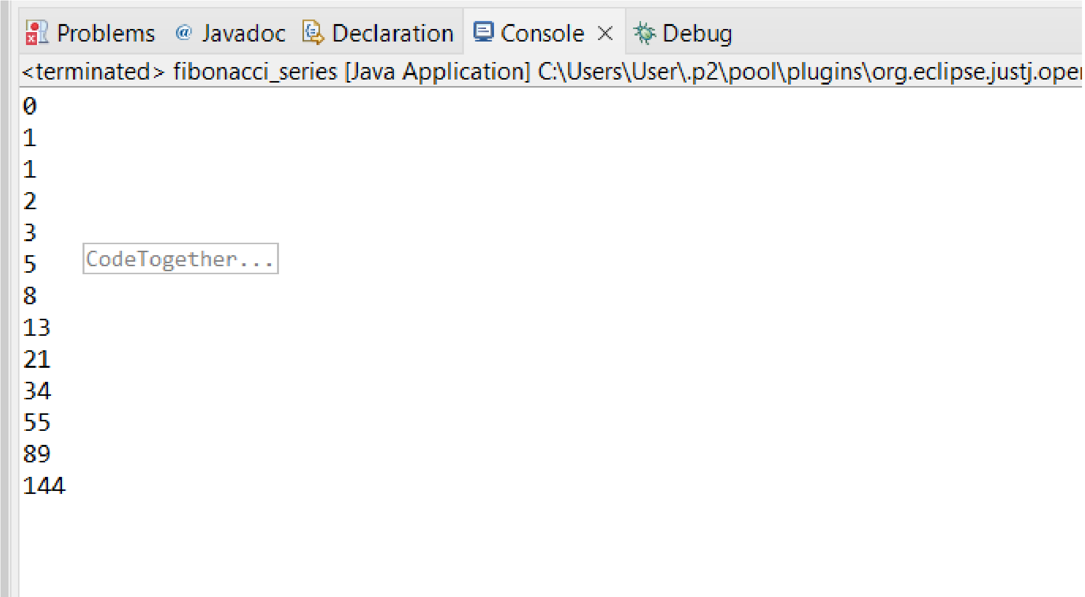


```
<terminated> Even_odd [Java Application] C:\Users\User\.p2\pool\plugin
Even
```

fibonacci_series:

```
public class fibonacci_series {

	static void fib(int i) {
		int num=0;
		int num1=1;
		System.out.println(num);
		System.out.println(num1);

		for(int j=0;j<=i;j++) {
			int num2=num+num1;
			num=num1;
			num1=num2;
			System.out.println(num2);
		}
	}
	public static void main(String[] args) {
		fib(10);

	}

}
```

Output:

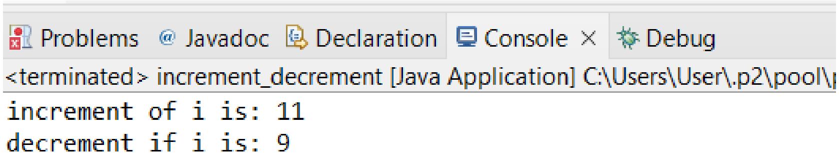


```
<terminated> fibonacci_series [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.ope
0
1
1
2
3
5
8
13
21
34
55
89
144
```

increment_decrement:

```
public class increment_decrement {

        static void inc(int i) {
                int j=++i;
                System.out.println("increment of i is: "+j);
        }
        static void dec(int i) {
                System.out.println("decrement if i is: "+--i);
        }
        public static void main(String[] args) {
                inc(10);
                dec(10);
        }

}
```

Output:

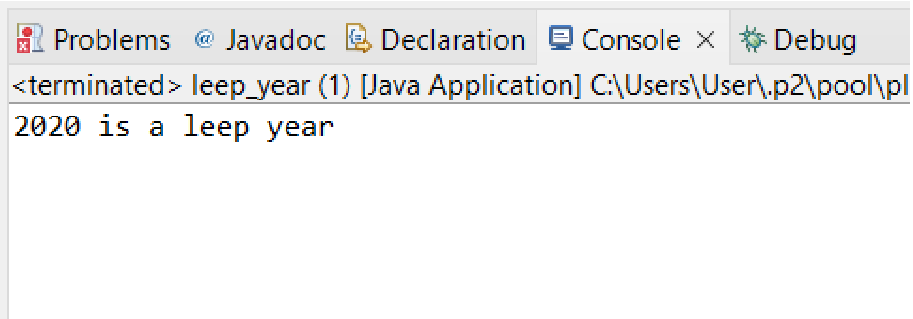


```
<terminated> increment_decrement [Java Application] C:\Users\User\.p2\pool\
increment of i is: 11
decrement if i is: 9
```

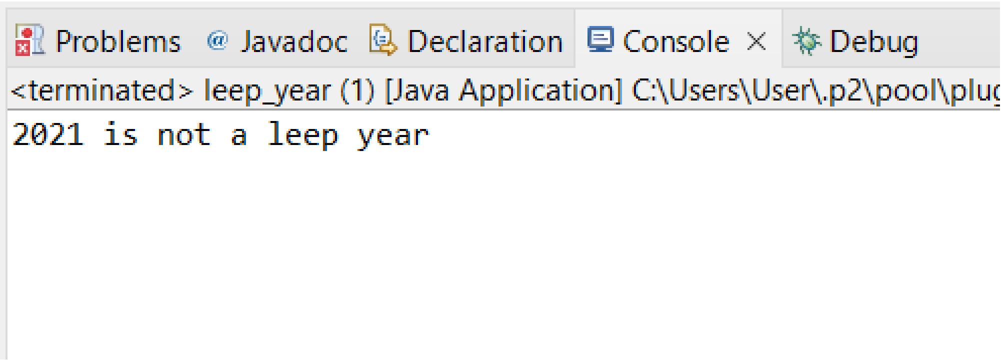
leep_year:

```java
public class leep_year {

	static void year(int year) {
		if(((year%4==0)&&(year%100!=0))||(year%400==0))
{System.out.println(year+" is a leep year");}
		else {System.out.println(year+" is not a leep year");}
	}
	public static void main(String[] args) {
		year(2020);

	}

}
```

Output:



```
<terminated> leep_year (1) [Java Application] C:\Users\User\.p2\pool\pl
2020 is a leep year
```

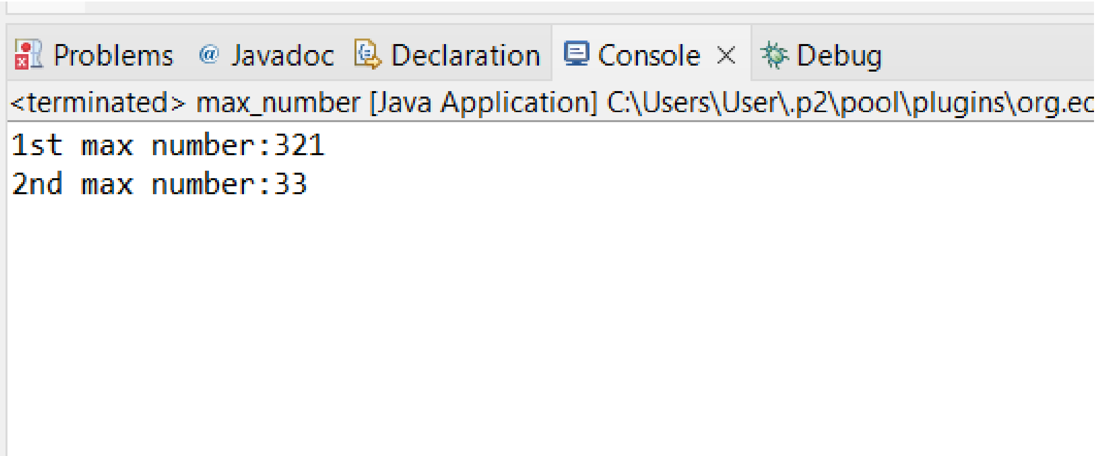


```
<terminated> leep_year (1) [Java Application] C:\Users\User\.p2\pool\plug
2021 is not a leep year
```

max_number:

```
    public class max_number {

      static void max(int array[]) {
            for(int i=0;i<array.length;i++) {
                  for(int j=i+1;j<array.length;j++) {
                        int temp=0;
                        if(array[j]< array[i]) {
                              temp=array[i];
                              array[i]=array[j];
                              array[j]=temp;
                        }
                  }
            }
            System.out.println("1st max number:"+array[array.length-
1]);
            System.out.println("2nd max number:"+array[array.length-
2]);}
      public static void main(String[] args) {

            int arr[]= {1,2,321,1,33,4};
            max(arr);
      }

}
```

Output:



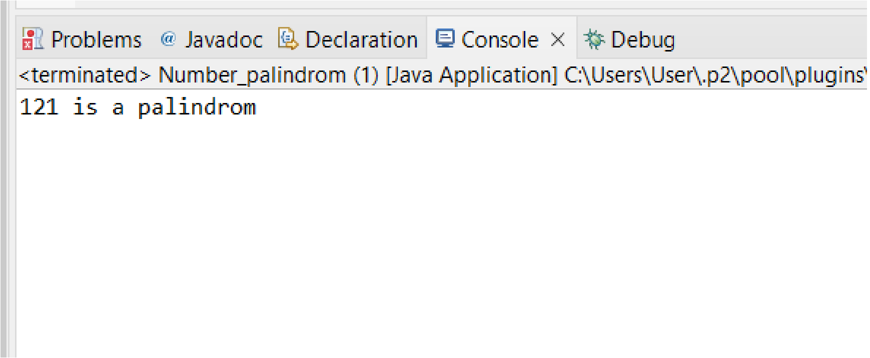
<terminated> max_number [Java Application] C:\Users\User\.p2\pool\plugins\org.ec

```
1st max number:321
2nd max number:33
```

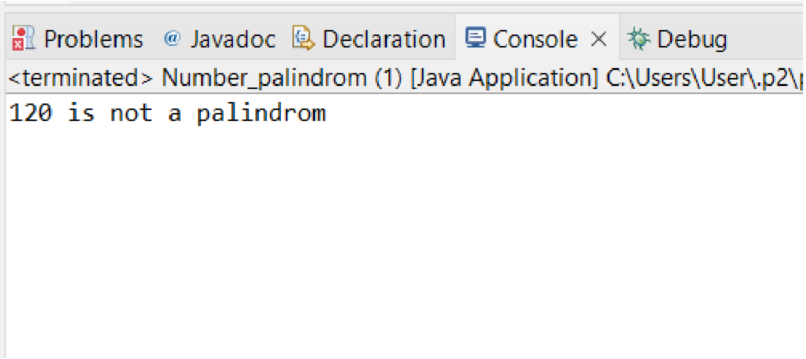
Number_palindrom:

```
    public class Number_palindrom {
        static void palindrom(int num1) {
        int num3=num1;
        int num2=0;
        while(num1!=0) {
            num2=num2*10+num1%10;
            num1=num1/10;
        }
        if(num2==num3) System.out.println(num3+" is a palindrom");
        else System.out.println(num3+" is not a palindrom");

    }
    public static void main(String[] args) {
        palindrom(121);

    }

}
```

Output:



```
<terminated> Number_palindrom (1) [Java Application] C:\Users\User\.p2\pool\plugins\
121 is a palindrom
```

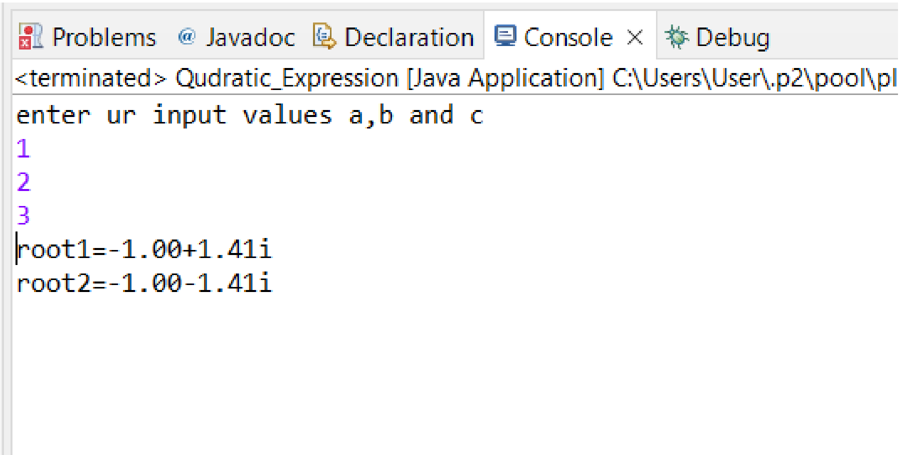


```
<terminated> Number_palindrom (1) [Java Application] C:\Users\User\.p2\
120 is not a palindrom
```

Qudratic_Expression:

```
    public class Qudratic_Expression {

    static void quadratic_expression(){

        Scanner sc=new Scanner(System.in);
        System.out.println("enter ur input values a,b and c");
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=sc.nextInt();
        double determinant=b*b-4*a*c;
        double root1 = 0,root2 = 0;
        //checking if determinant is greater than 0.
        if(determinant>0)
        {
            root1=(-b+Math.sqrt(determinant))/(2*a);
            root2=(-b+Math.sqrt(determinant))/(2*a);
        System.out.format("root1 =%.2f and  root2=%.2f", root1,root2);
        }
        //checking if determinant is equal to 0
        else if(determinant==0)
        {
        root1=root2=-b/(2*a);
        System.out.format("root1=root2=%.2f",root1);
        }
        else
            //if determinant is less than 0.
        {
            //roots are complex number and distinct.
            double real=-b/(2*a);
            double imaginary=Math.sqrt(-determinant)/(2*a);
            System.out.format("root1=%.2f+%.2fi", real,imaginary);
            System.out.format("\nroot2=%.2f-%.2fi",real,imaginary);
        }

    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        quadratic_expression();
    }

}
```

Output:



```
<terminated> Qudratic_Expression [Java Application] C:\Users\User\.p2\pool\pl
enter ur input values a,b and c
1
2
3
root1=-1.00+1.41i
root2=-1.00-1.41i
```

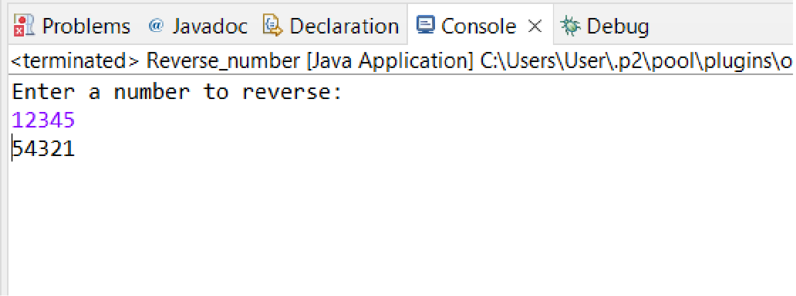
Radix:

```
    public class Radix {

    static void radix() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter u r format:");
        String format=sc.nextLine();
        if(format.matches("[0-1]+")) {
            System.out.println("you Entered Binary values with Base 2");
        }
        else {
            if(format.matches("[0-7]+")) {
                System.out.println("you Entered Octal values with Base 
8");
            }
            else {
                if(format.matches("[0-9]+")) {
                    System.out.println("you Entered Decimal values 
with Base 10");
                }
                else if(format.matches("[O-9A-F]+")) {
                    System.out.println("you Entered Hexadecimal value 
with Base 16");
                }
                else {System.out.println("wrong format please enter 
correct fromat");}
            }
        }
    }
     public static void main(String[] args) {
         // TODO Auto-generated method stub
        radix();
    }

}
```

Reverse_number:

```
public class Reverse_number {

	static void reverse() {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter a number to reverse:");
		int num=sc.nextInt();

		while(num!=0) {
			int num1=num%10;
			System.out.print(num1);
			num=num/10;
		}
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		reverse();
	}

}
```

Output:

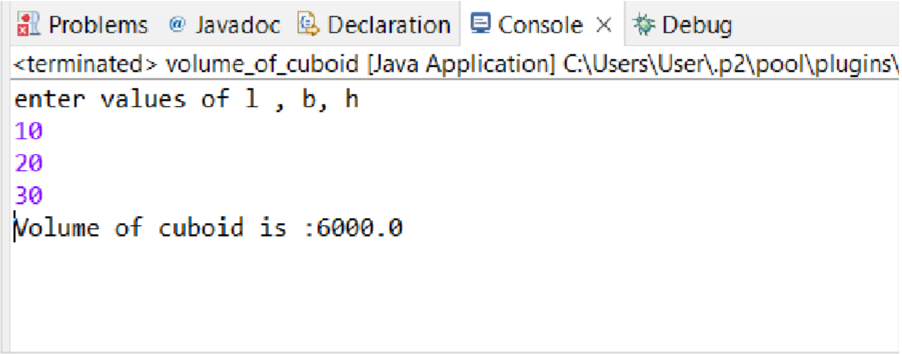


```
<terminated> Reverse_number [Java Application] C:\Users\User\.p2\pool\plugins\o
Enter a number to reverse:
12345
54321
```

volume_of_cuboid:

```
public class volume_of_cuboid {

	static	void	check_volume_of_cuboid(){
		Scanner sc=new Scanner(System.in);
		System.out.println("enter values of l , b, h");
		int l=sc.nextInt();
		int b=sc.nextInt();
		int h=sc.nextInt();

		double volume=l*b*h;
		System.out.println("Volume of cuboid is :"+volume);
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		check_volume_of_cuboid();
	}
	}
```

Output:



```
<terminated> volume_of_cuboid [Java Application] C:\Users\User\.p2\pool\plugins\
enter values of l , b, h
10
20
30
Volume of cuboid is :6000.0
```