1.

a) i. 3
ii. 5
iii. 0

b)
mystery method is computing int(a/b)

2. a)

i) value returned :: -72
methods called :: mystery2(7,2) => -mystery2(7,-(-2)) => 100 * mystery2(0,0) + 10*(7) + 2

ii) value returned :: 2495
methods called :: mystery2(29,45) => 100 * mystery2(2,4) + 10*(9) + 5 => 100 * mystery2(0,0) + 10*(2) + 4

iii) value returned :: 123456
methods called :: mystery2(135,246) => 100 * mystery2(13,24) + 10*(5) + 6 => 100 * mystery2(1,1) + 10*(3) + 4 => 100 * mystery2(0,0) + 10*(1) + 2

b)

mystery method is returning digits from left to right and alternatively between two numbers X and Y

3. a)

```
int power(m, n)
   if (m<0)
        return power(-m,n)
   else if (n==1)
        return m
   else
        return m * power(m, n-1)
```

b)

I)

```
int power(m, n)
if (m<0)
        return power(-m,n)
else if (n==1)
        return m
else
        if(n%2==1)
```

```
                return power(m, n/2)* power(m, n/2)
        else
                return m * power(m, n/1)* power(m, n-1)
```

II)

11 recursive calls will occur when computing power(x, 1024)

4

```
public class Main
        {
                public static int rev = 0;

 static int reverse_number(int n)
                {

                if (n == 0)
                          return 0;

 rev = rev * 10;
       rev = rev + n % 10;
                        reverse_number(n/10);
                         return rev;
                }

                public static void main(String[] args)
                {
                    int reverse = reverse_number(123345);
                        System.out.println(reverse);
                }
}
```