Assignment No. 2

```
1.class array{
        public static void main(String[]args){
                int[]arr={1,2,3,4};
                printArray(arr,0);
        }
        static void printArray(int[] arr,int s){
                if(s>=arr.length){
                        return;}

                System.out.println(arr[s]);
                printArray(arr,s+1);
                System.out.println(arr[s]);


        }
}
```

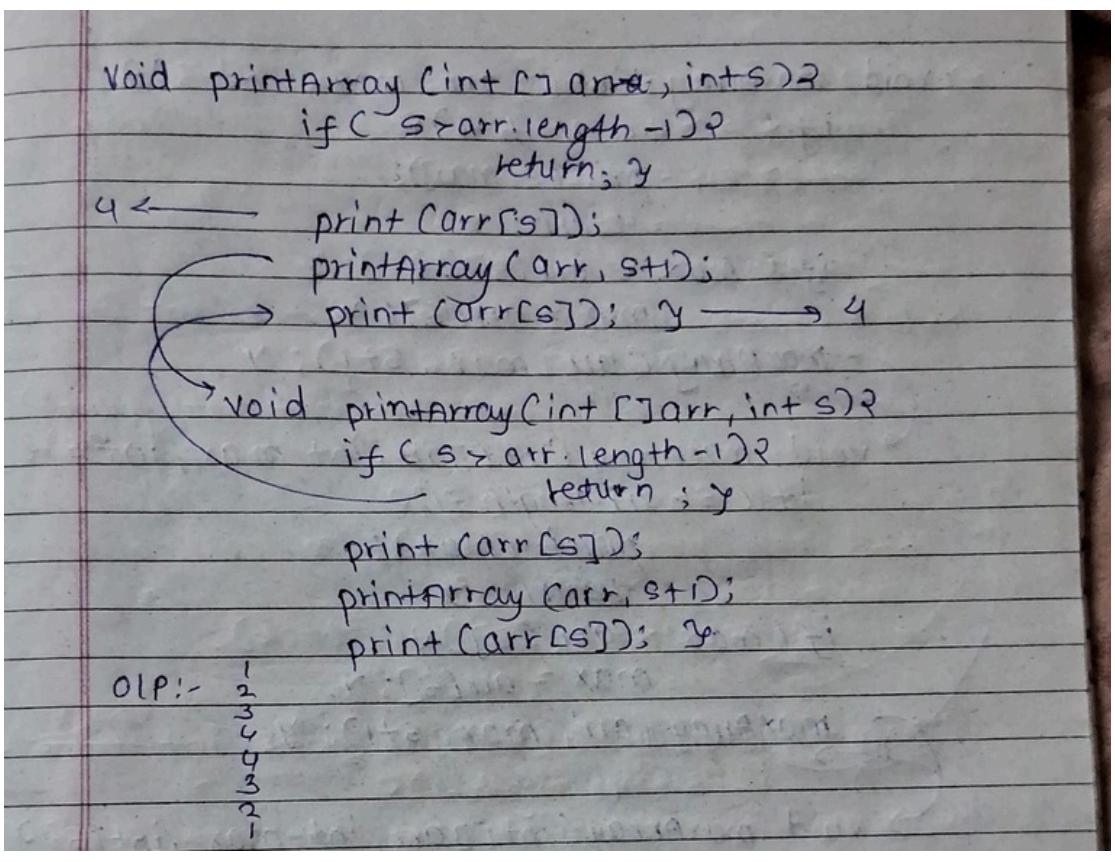
4) print elements of an array (forward & backward)

```
main()
int[] arr = [1,2,3,4]
printArray(arr,0);
void printArray(int[] arr, int s){
    if(s > arr.length-1){
        return;
    print(arr[s]);
    printArray(arr, s+1);
    print(arr[s]);
}
```

```
void printArray(int[] arr, int s){
    if(s > arr.length-1){
        return;
    print(arr[s]);
    printArray(arr, s+1);
    print(arr[s]);
}
```

```
void printArray(int[] arr, int s){
    if(s > arr.length-1){
        return;
    print(arr[s]);
    printArray(arr, s+1);
    print(arr[s]);
}
```

```
void printArray(int[] arr, int s){
    if(s > arr.length-1){
        return;
    print(arr[s]);
    printArray(arr, s+1);
    print(arr[s]);
}
```



O/P:-

1
2
3
4
4
3
2
1

```
2.class sum{
public static void main(String[]args){
        int number=1234;
        sumOfDigits(number,0);
}
    static void sumOfDigits(int number,int sum){
        if(number==0){
            System.out.println(sum);
            return; }
        int remainder=number%10;
        sum=sum+remainder;
        number=number/10;
        sumOfDigits(number,sum);
}
}
```

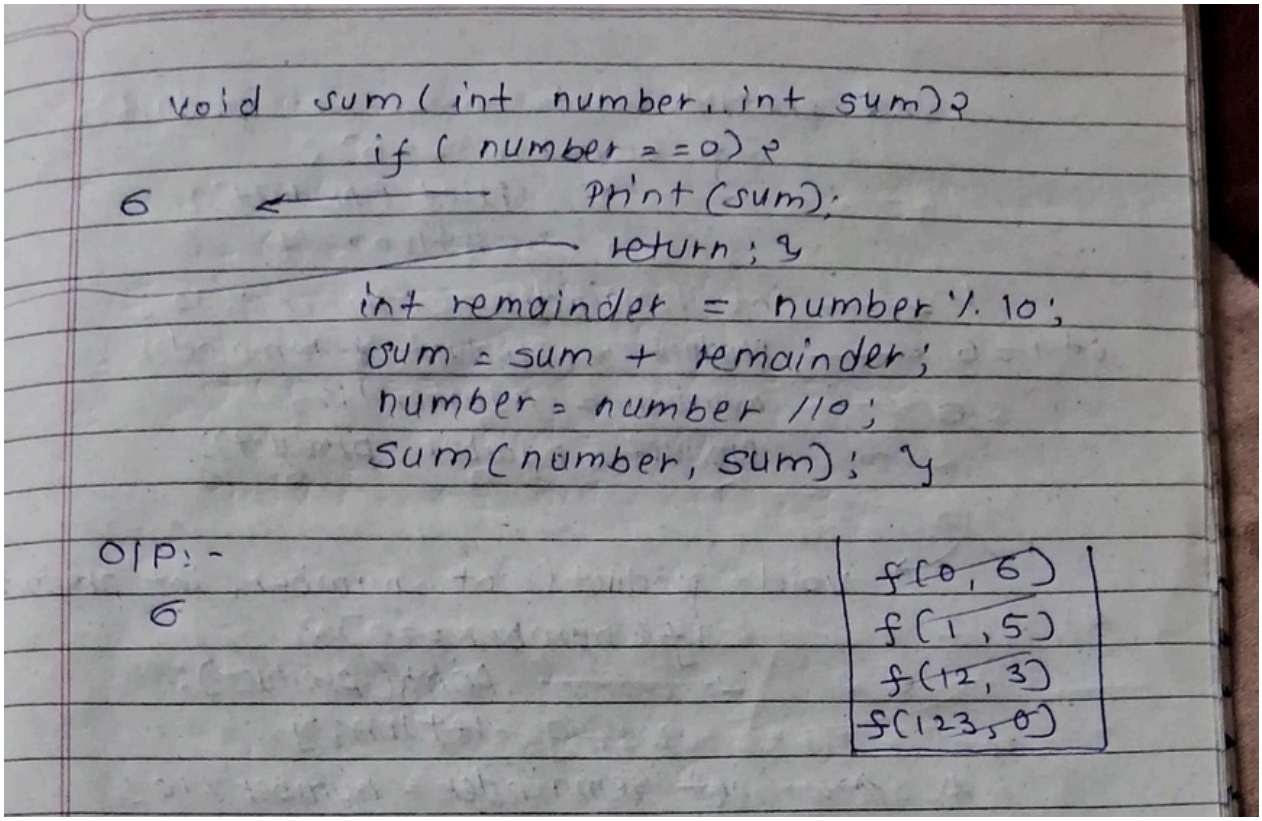
Q.1. find the sum of digits ...

```
main()
    int number = 123;
    sum(number, 0);

void sum(int number, int sum)
    if (number == 0)
        print(sum);
        return;
    int remainder = number % 10;
    sum = sum + remainder;
    number = number / 10;
    sum(number, sum);

void sum(number, int sum)
    if (number == 0)
        print(sum);
        return;
    int remainder = number % 10;
    sum = sum + remainder;
    number = number / 10;
    sum(number, sum);

void sum(int number, int sum)
    if (number == 0)
        print(sum);
        return;
    int remainder = number % 10;
    sum = sum + remainder;
    number = number / 10;
    sum(number, sum);
```

3 ← int remainder = number % 10;
3+3 ← sum = sum + remainder;
2 ← number = number / 10;
sum(number, sum);
12 3

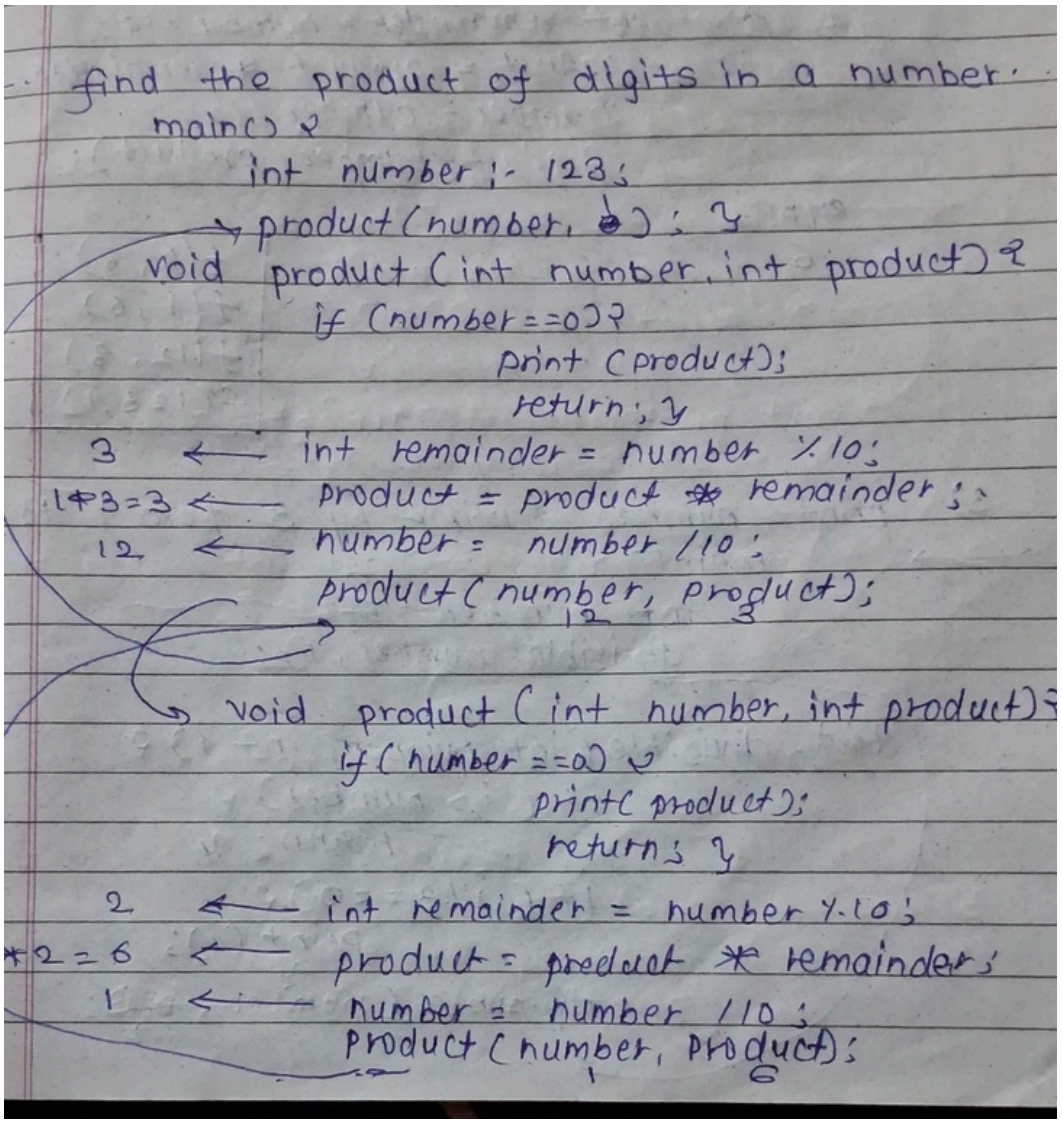
2 ← int remainder = number % 10;
3+2=5 ← sum = sum + remainder;
1 ← number = number / 10;
sum(number, sum);
1 5

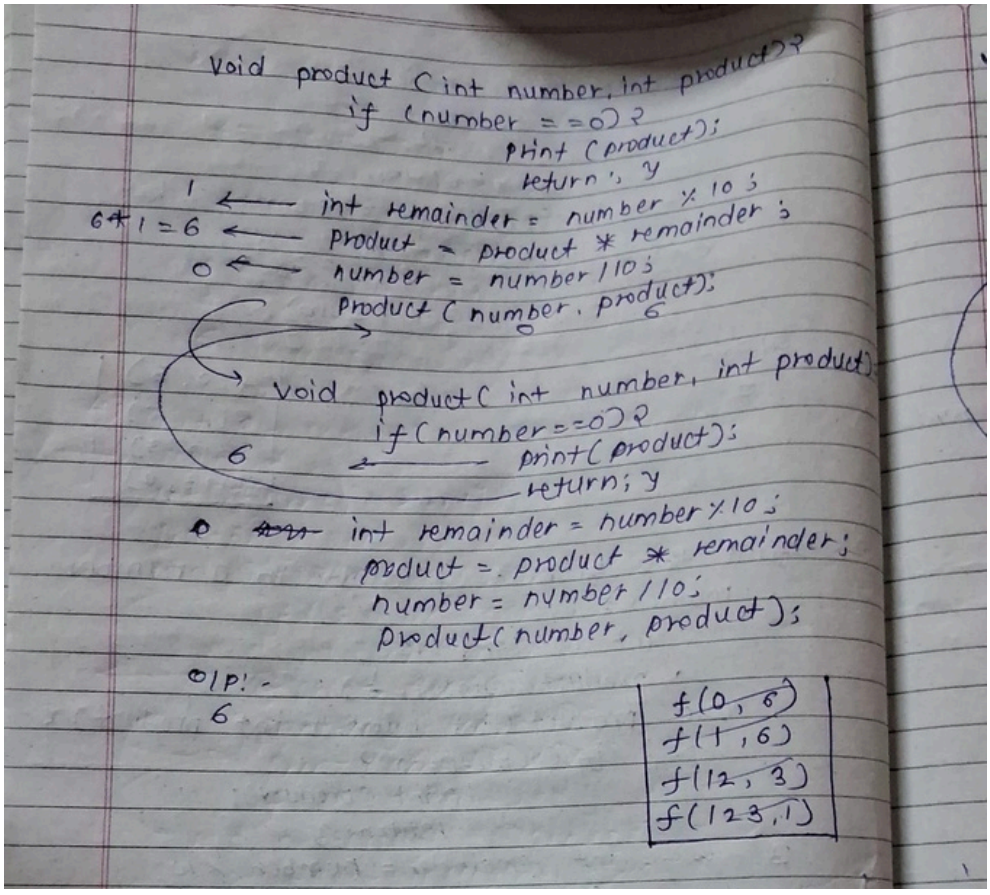
1 ← int remainder = number % 10;
5+1=6 ← sum = sum + remainder;
0 ← number = number / 10;
sum(number, sum);
0 6




```
3.class product{
public static void main(String[]args){
        int number=1234;
        productOfDigits(number,1);
}
    static void productOfDigits(int number,int product){
        if(number==0){
            System.out.println(product);
            return; }
        int remainder=number%10;
        product=product*remainder;
```

```
number=number/10;
productOfDigits(number,product);
}}
```

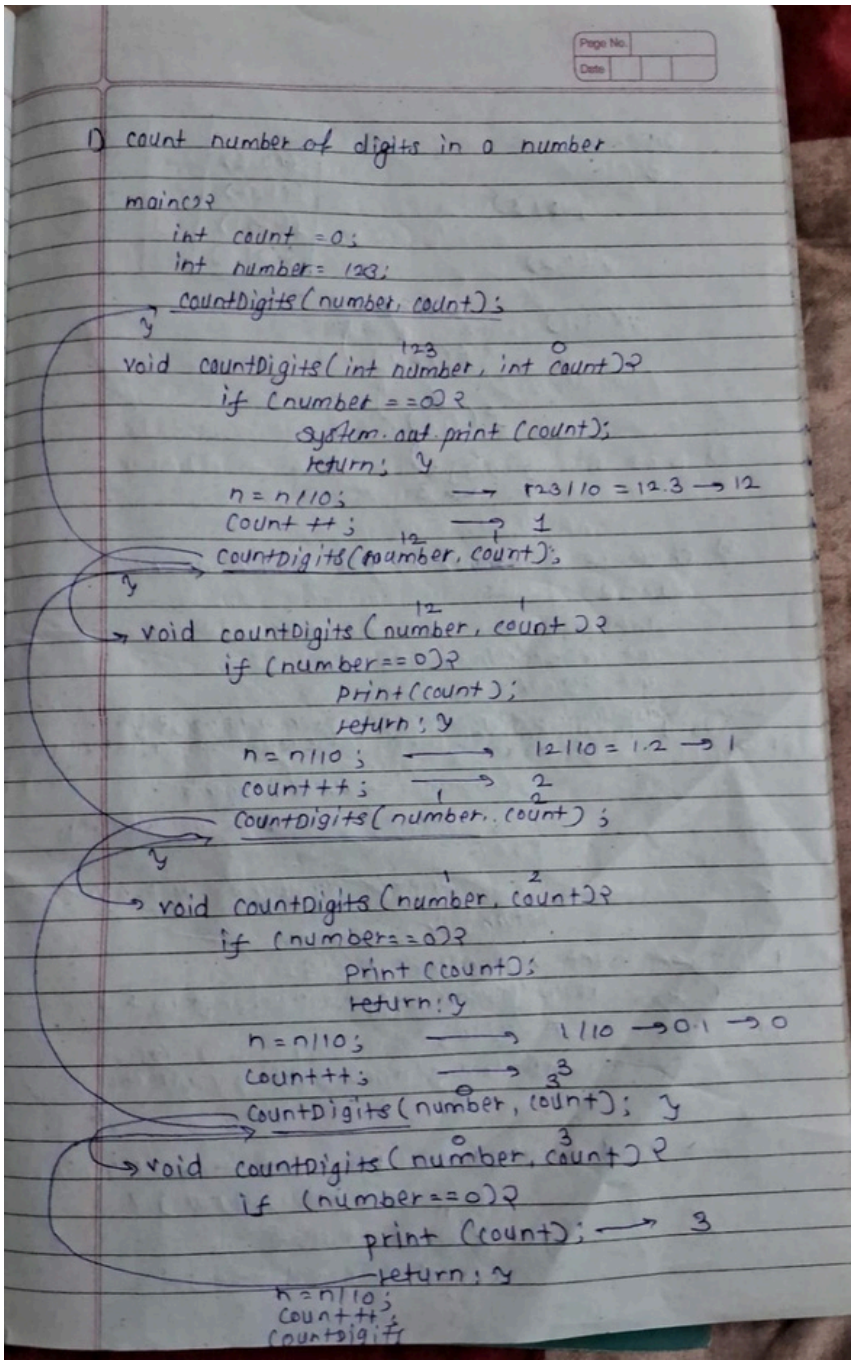


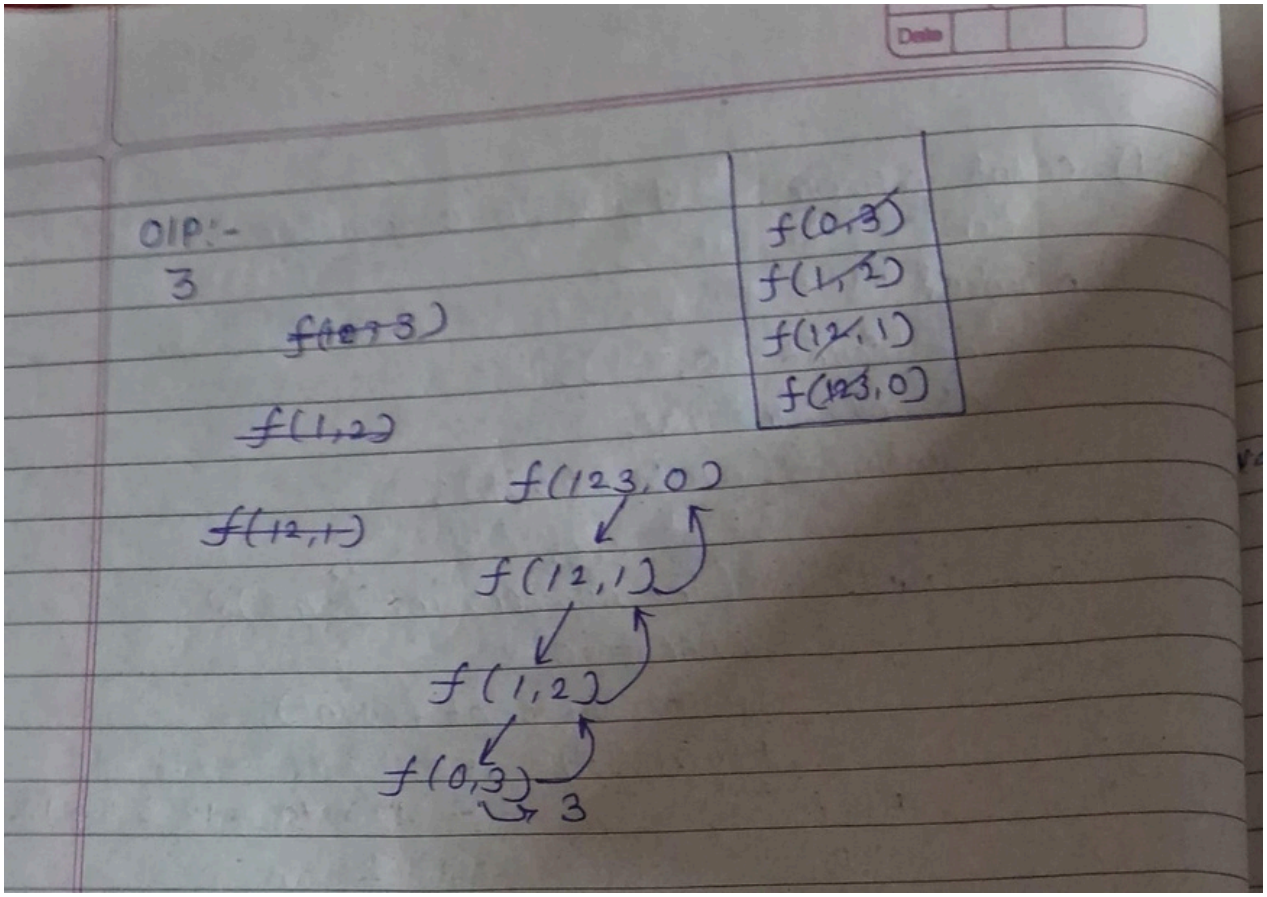


4.

```
class coundigits{
    public static void main(String[]args){
        int n=12345;
        int count=0;
        f(count,n);
        }
    static void f(int count,int n){
        if(n==0){
            System.out.println(count);
            Return;
}
```

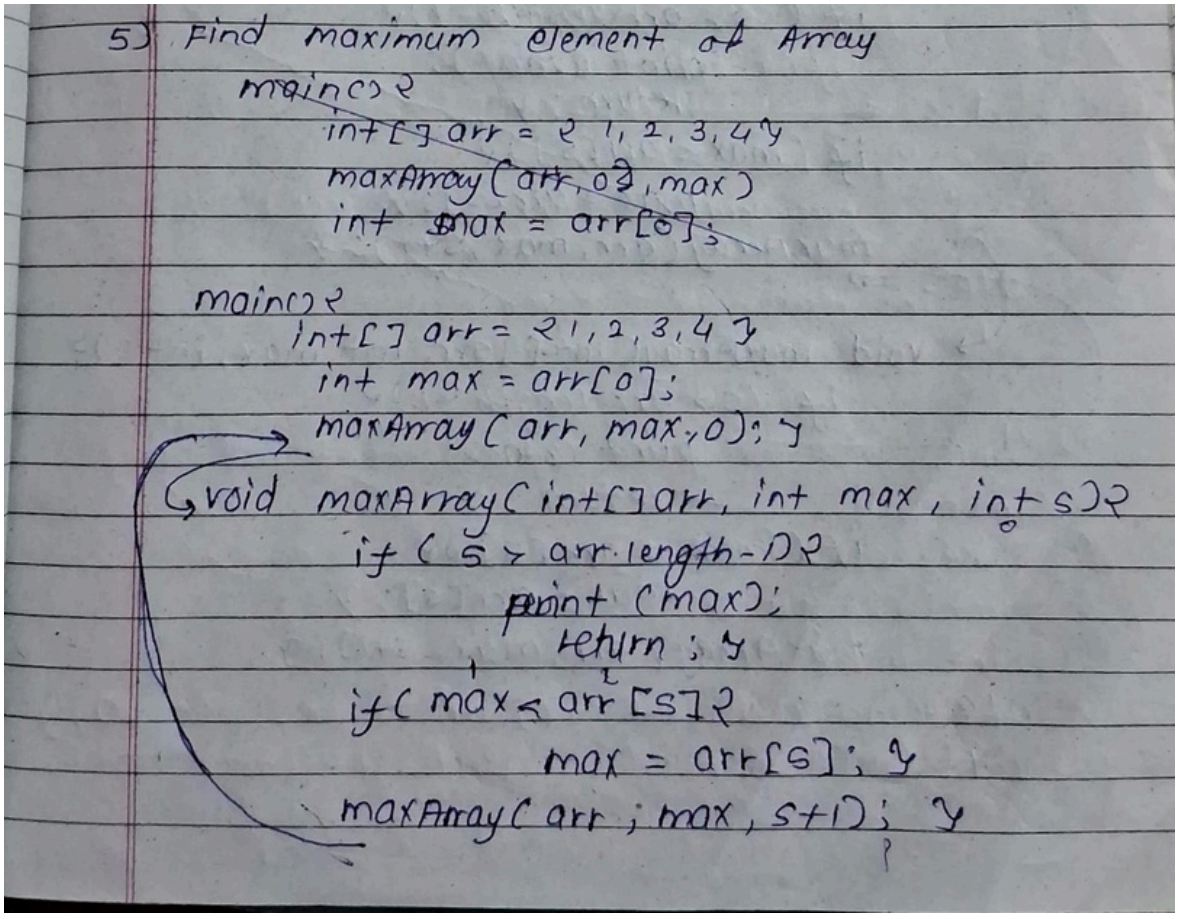
```
        n=n/10;
        count++;
        f(count,n);}
}
```

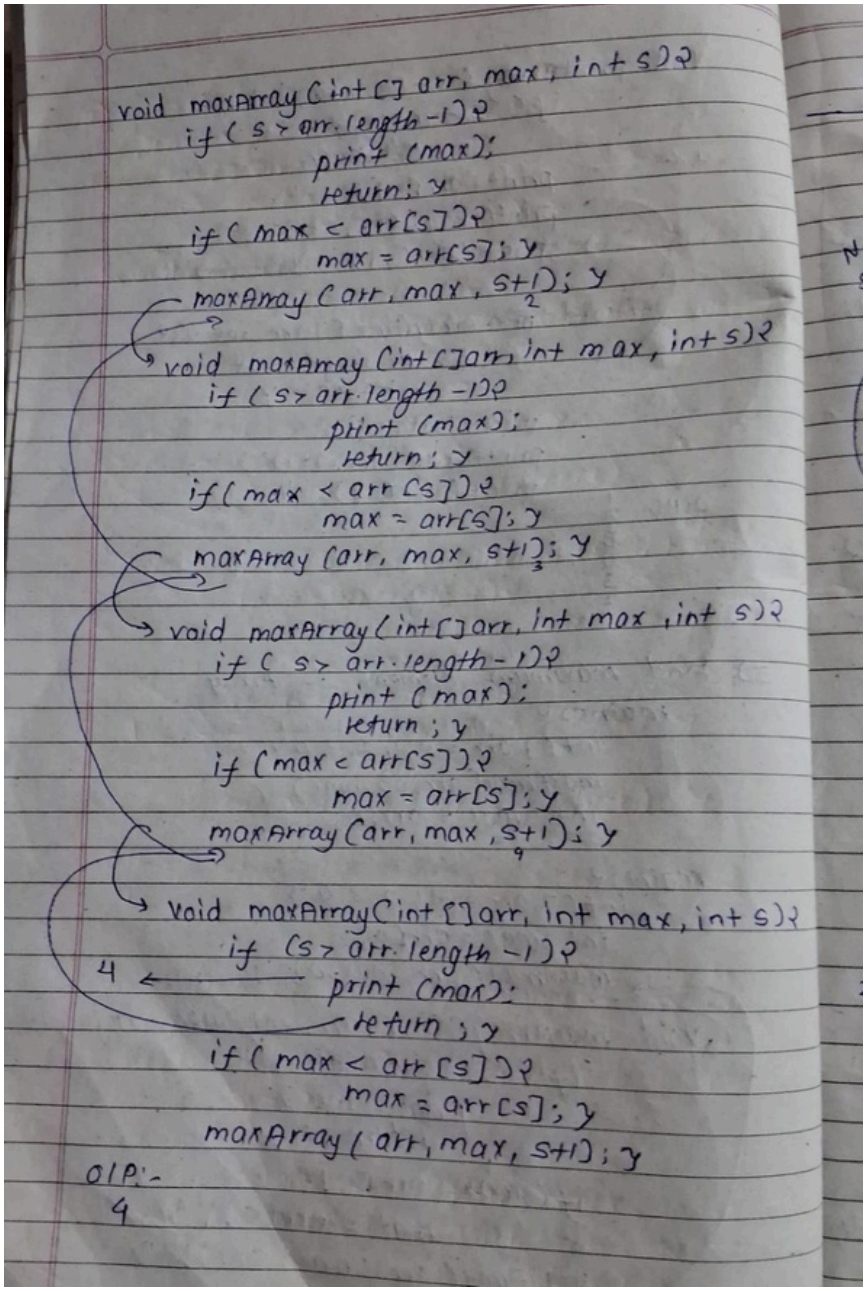





```
5. class array1{
    public static void main(String[]args){
        int[]arr={1,5,3,4};
        int s=0;
        int max=arr[s];
        maxArray(arr,s,max);
}
    static void maxArray(int[] arr,int s,int max){
        if(s>=arr.length){
        System.out.println(max);
        return;}
        if(max <arr[s]){
```

```
            max=arr[s];}
        maxArray(arr,s+1,max);

}
}
```





6. class sortedArray{

public static void main(String[]args){

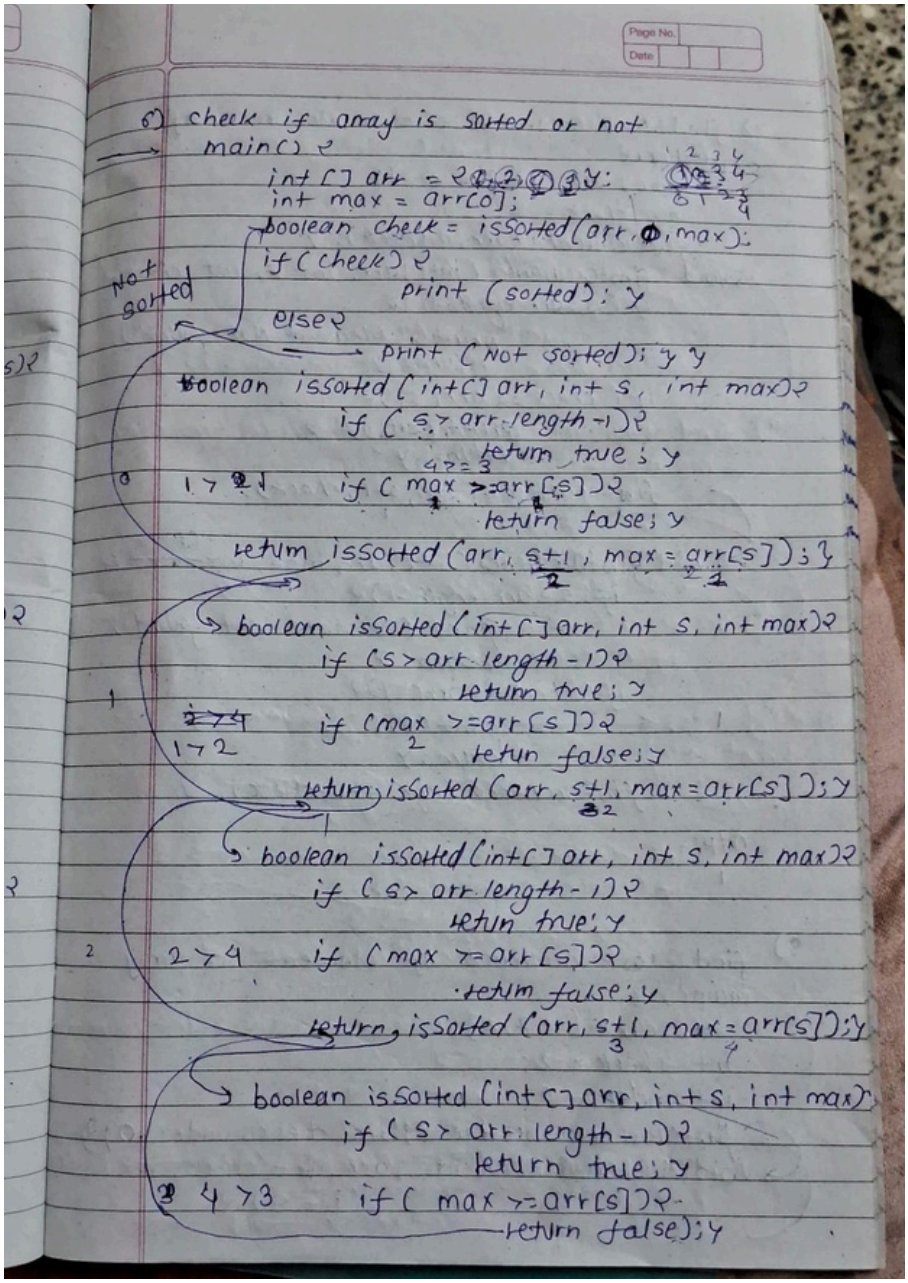
int[]arr={1,3,3,4};

int s=0;

int max=arr[s];

boolean check=sortedArray(arr,s,max);

```
            if(check){
                System.out.println("Sorted");}
            else{
                System.out.println(" not Sorted");}
}
    static boolean sortedArray(int[] arr,int s,int
max){
            if(s>=arr.length-1){
            return true;
            }
             if(max >=arr[s]){
                 return false;}

            return sortedArray(arr,s+1,max=arr[s]);

    }
}
```



7.class prime{

public static void main(String[]args){

int number=1;

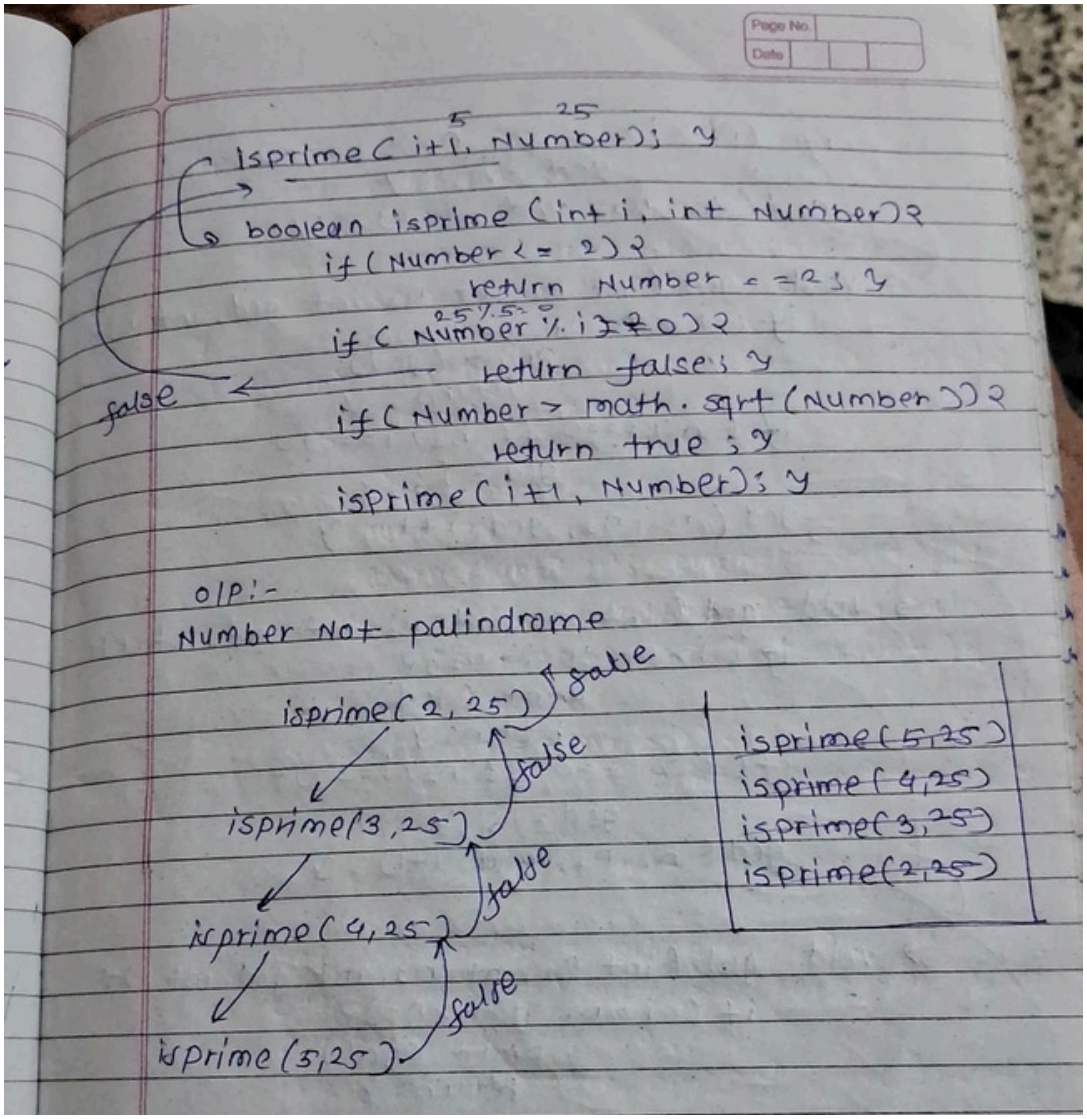
if(isPrime(number,2)){

System.out.println("Number is prime");

```
            }
        else{
            System.out.println("Number is not
prime");}
}
        static Boolean isPrime(int number,int i){
            if(number<=2){
                return number==2;}
            if(number%i==0){
                return false;}
            if(number>Math.sqrt(number)){
                return true;}
            return isPrime(number,i++);}
}
```

5) check if a Number is prime or not.

```
main() {
        int Number = 25;
boolean check = isPrime(2, Number);
 false  if (check)
            print(" Number is prime"); }
        else {
            print(" Number not prime"); }
}
boolean isPrime(int 2, int 25 Number) {
        if (Number <= 2) {
                return number==2; }
        if (Number % i == 0) {
                return false; }
        if (Number > math.sqrt(Number)) {
                return true; }
        isPrime(i+1, Number);
                 3   25
}

boolean isPrime(int i, int Number) {
        if (Number <= 2) {
                return Number==2; }
        if (Number % i == 0) {
                return false; }
        if (Number > math.sqrt(Number)) {
                return true; }
        isPrime(i+1, Number);
                 4   25

boolean isPrime(int i, int Number) {
        if (Number <= 2) {
                return Number == 2; }
        if (Number % i == 0) {
                return false; }
        if (Number > math.sqrt(Number)) {
                return true; }
```



8. class FirstOccurence{

public static void main(String[]args){

int[]arr={1,2,2,4};

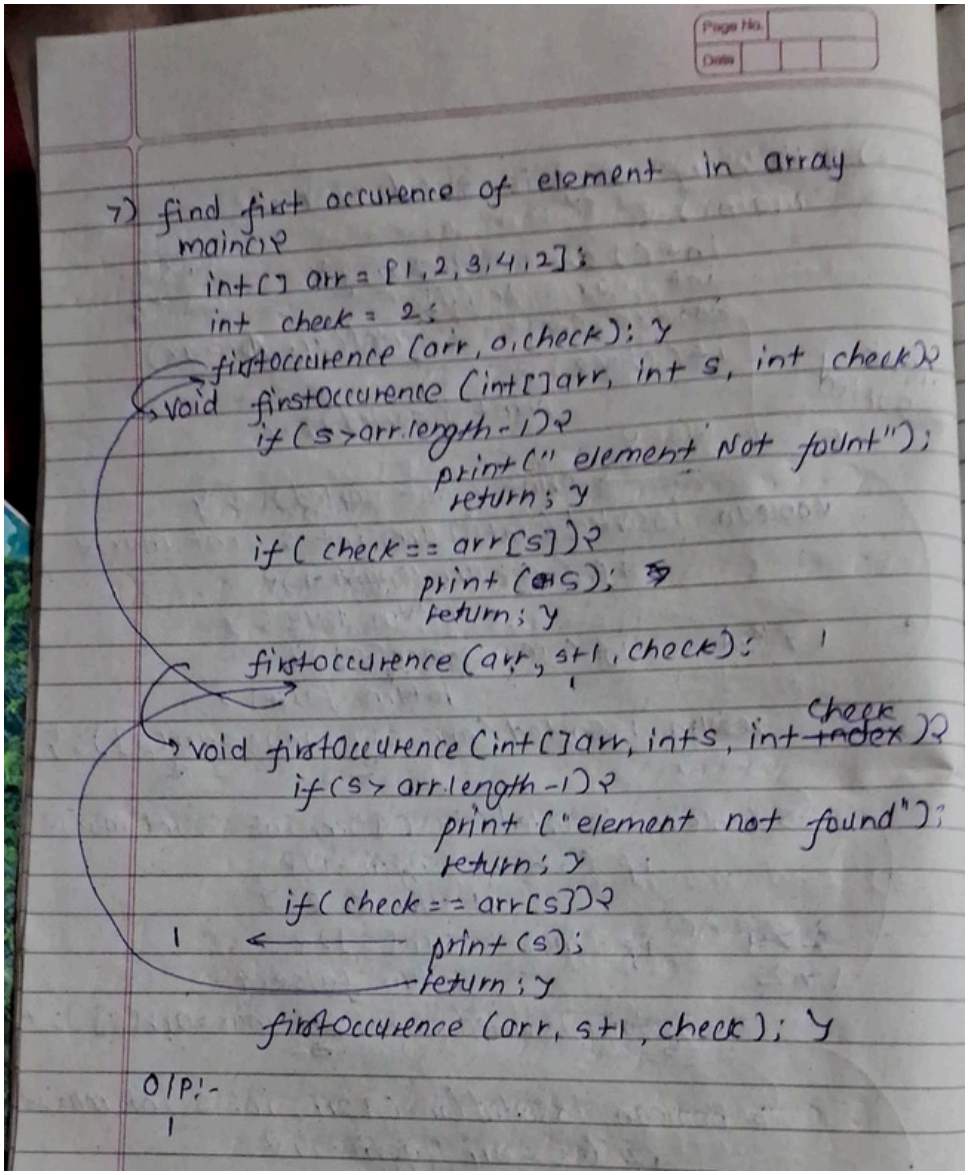
int number=2;

FirstOccurence(arr,0,number);

}

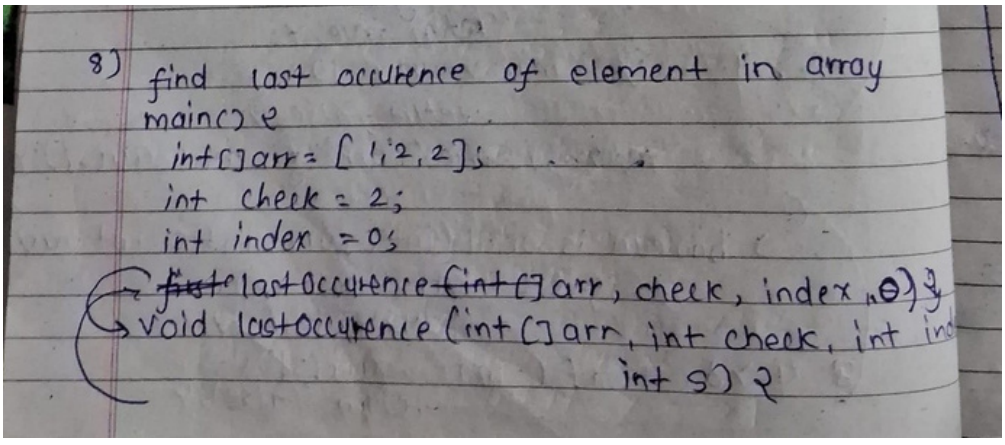
static void FirstOccurence(int[] arr,int s,int number){

```
            if(s>=arr.length-1){
                System.out.println("Element not
found");
                return;
                }

            if(number==arr[s]){
                System.out.println(s);
                return ;}
             FirstOccurence(arr,s+1,number);
            }
}
```

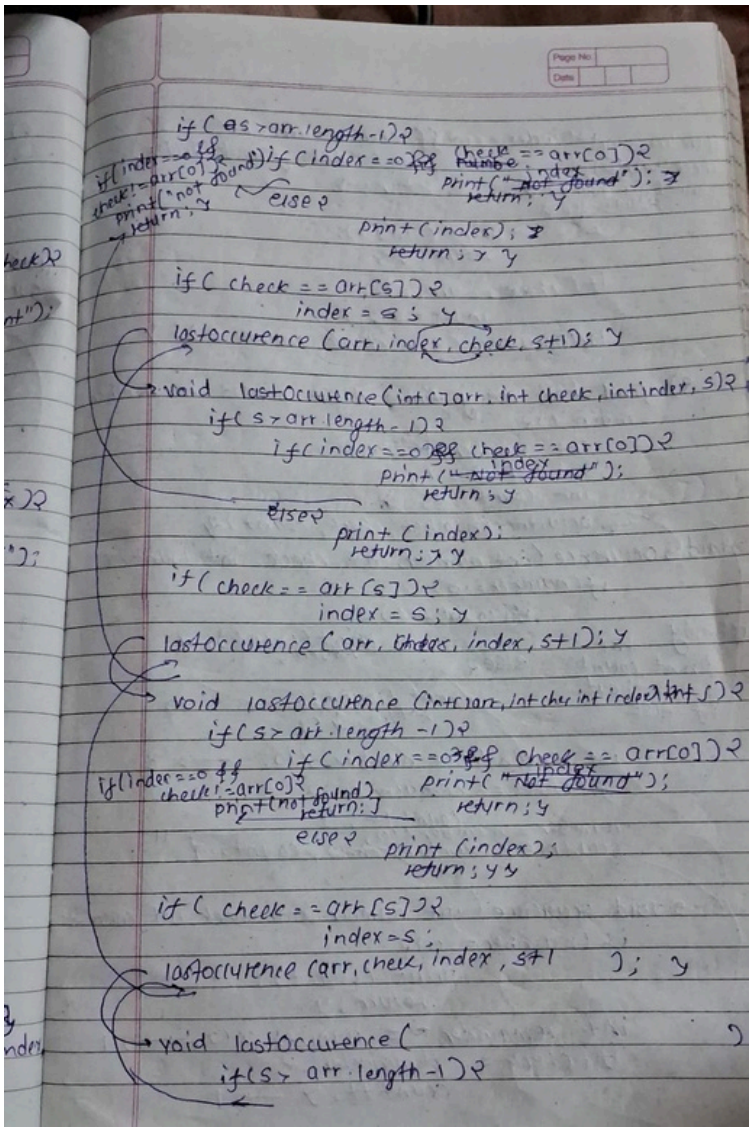


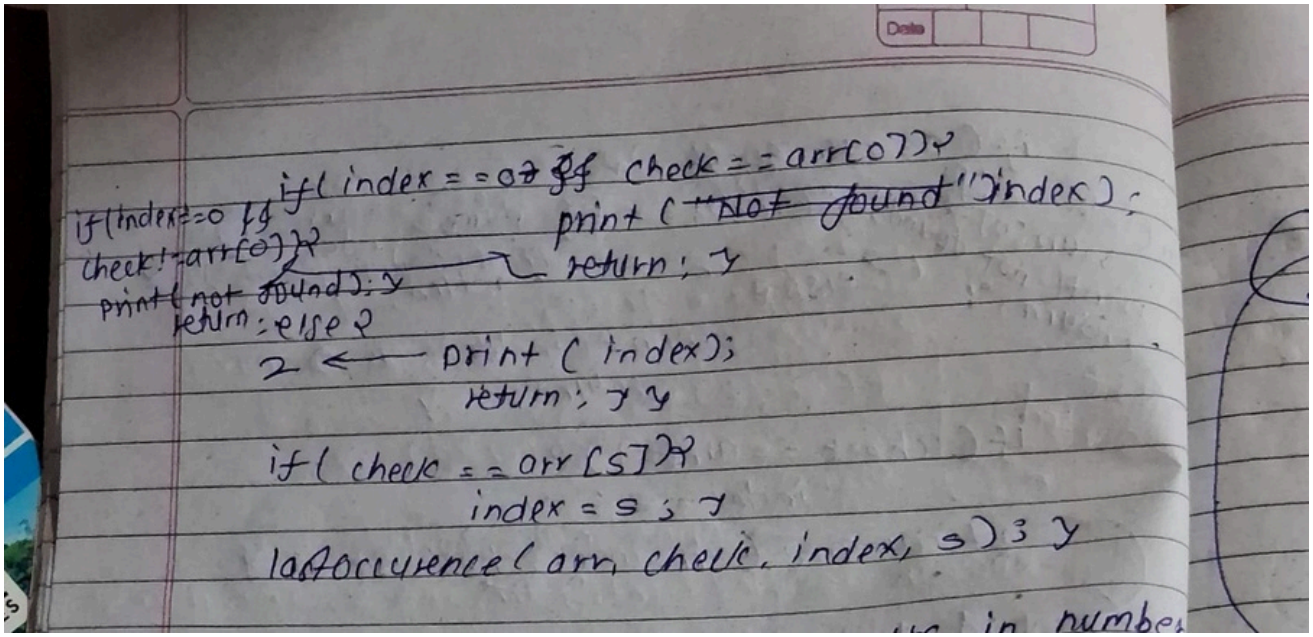

```
9. class LastOccurence{
      public static void main(String[]args){
            int[]arr={1,5,8,4};
            int number=2;
            int index=0;
            FirstOccurence(arr,0,number,index);
}
```

```
    static void FirstOccurence(int[] arr,int s,int
number,int index){
            if(s>arr.length-1){
                 if(index==0 && number==arr[0]){
                      System.out.println(index);
                return;
}               if(index==0&&number!=arr[0]){
                      System.out.println("not found");
}else{
                      System.out.println(index);
                       return;
}}
            if(number==arr[s]){
                 index=s;
                 ;}
             FirstOccurence(arr,s+1,number,index);

}
```




```
}
```





10. class reverse{

public static void main(String[]args){

int number=12345;

int reversedNumber=0;

reverseTheNumber(number,reversedNumber);

}

public static void reverseTheNumber(int number,int reversedNumber){

if(number==0){

System.out.println(reversedNumber);

return;}

int remainder=number%10;

```
reversedNumber=reversedNumber*10+remainde
r;
                number=number/10;


reverseTheNumber(number,reversedNumber);
            }
}

11. class OccurenceinNumber{
      public static void main(String[]args){
            int number=14444;
            int check=2;
            int count=0;

      OccurenceinNumber(number,check,count);
}
      static void OccurenceinNumber(int
number,int check,int count){
            if(number==0){
                  if(count==0){
                        System.out.println(count);
            return;}
                  else{
                        System.out.println(count);
```

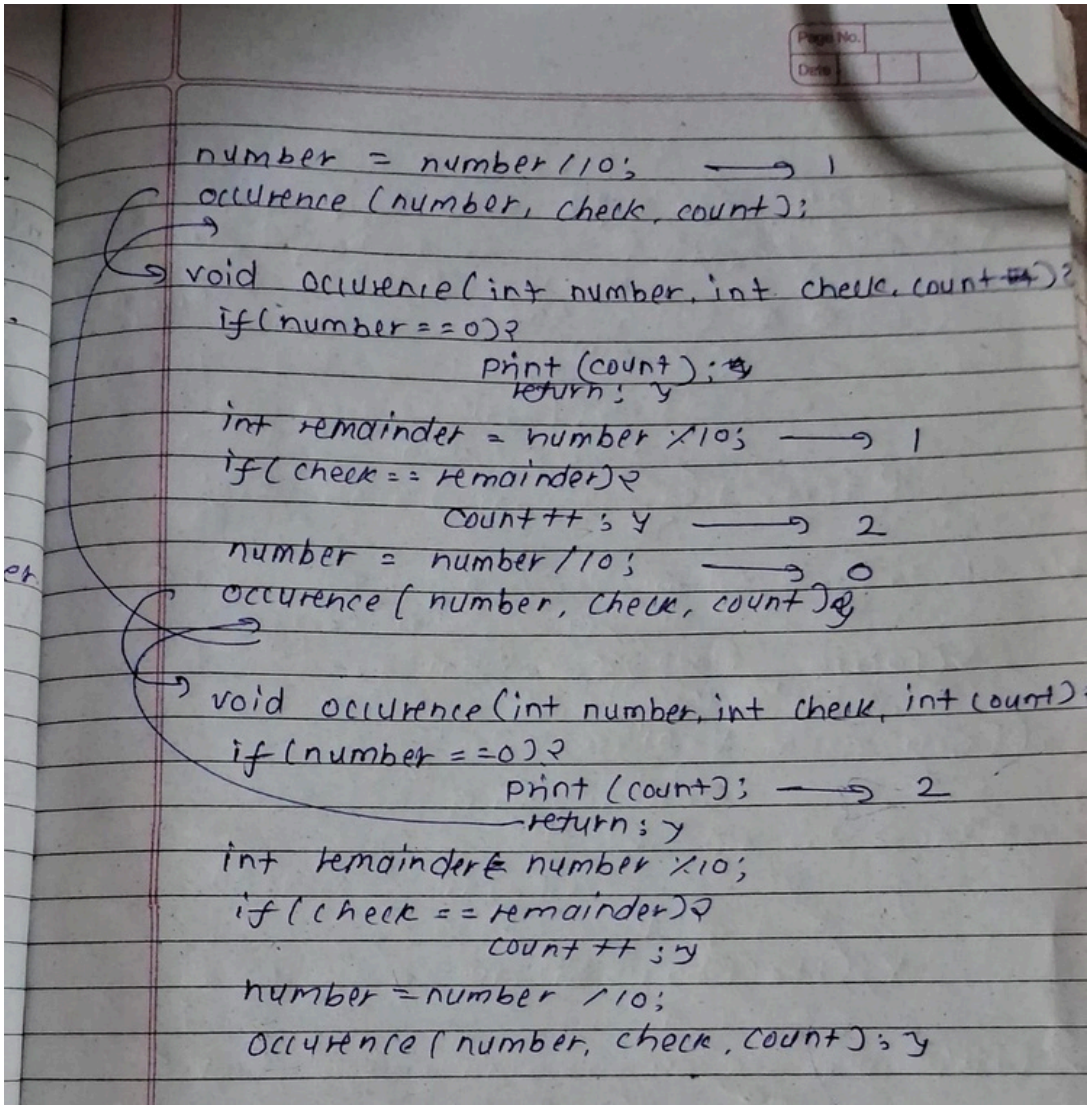
```
                    return;
}
}
        int remainder=number%10;
        if(check==remainder){
            count++;
            ;}
        number=number/10;

OccurenceinNumber(number,check,count);

    }
}
```

9) find how many times digit occurs in number

```
main ()
    int number = 113;
    int check = 1;
    int count = 0;
    occurence(number, check, count);
void occurence (int number, int check, int count)
    if(number==0)
        if(count==0)
            print(count); return;
        else
            print(count); return;
    int remainder = number %10;
    if(check == remainder)
        count++;
    number = number/10;
    occurence (number, check, count);
```

simply the print count without if-else

```
void occurence (int number, int check, int count)
    if (number==0)
        print (count); return;
    int remainder = number %10;
    if (check == remainder)
        count++;
```



12. class palindrome{

public static void main(String[]args){

int n=11;

int reversedNumber=0;

int original=n;

isPalindrome(reversedNumber,n,original);

}

```
static void isPalindrome(int reversedNumber,int n,int original){
                if(n==0){
                    if(original== reversedNumber){
                System.out.println("Number is Palindrome");}
                        else{
                System.out.println("Number is not Palindrome");}
            return;}
                int remainder=n%10;
                n=n/10;


reversedNumber=reversedNumber*10+remainder;

      isPalindrome(reversedNumber,n,original);}
}
```

2) check if a number is palindrome or not

```
main()
    int n = 121;
    int reversedNumber = 0;
    int original = n;
    ispalindrome(n, reversedNumber, original);
}
void ispalindrome(int n, int reversedNumber, int original)
{
    if(n==0)
    {
        if(original == reversedNumber)
            print("Number is palindrome");
        else
            print(" Number is Not palindrome");
        return;
    }
    int remainder = n%10;
    reversedNumber = reversedNumber * 10 + remainder;
    n = n/10;
    ispalindrome(n, reversedNumber, original);
}
```

121, 0, 121

1 ← int remainder = n%10;

0*10+1=1 ← reversedNumber = reversedNumber * 10 + remainder;

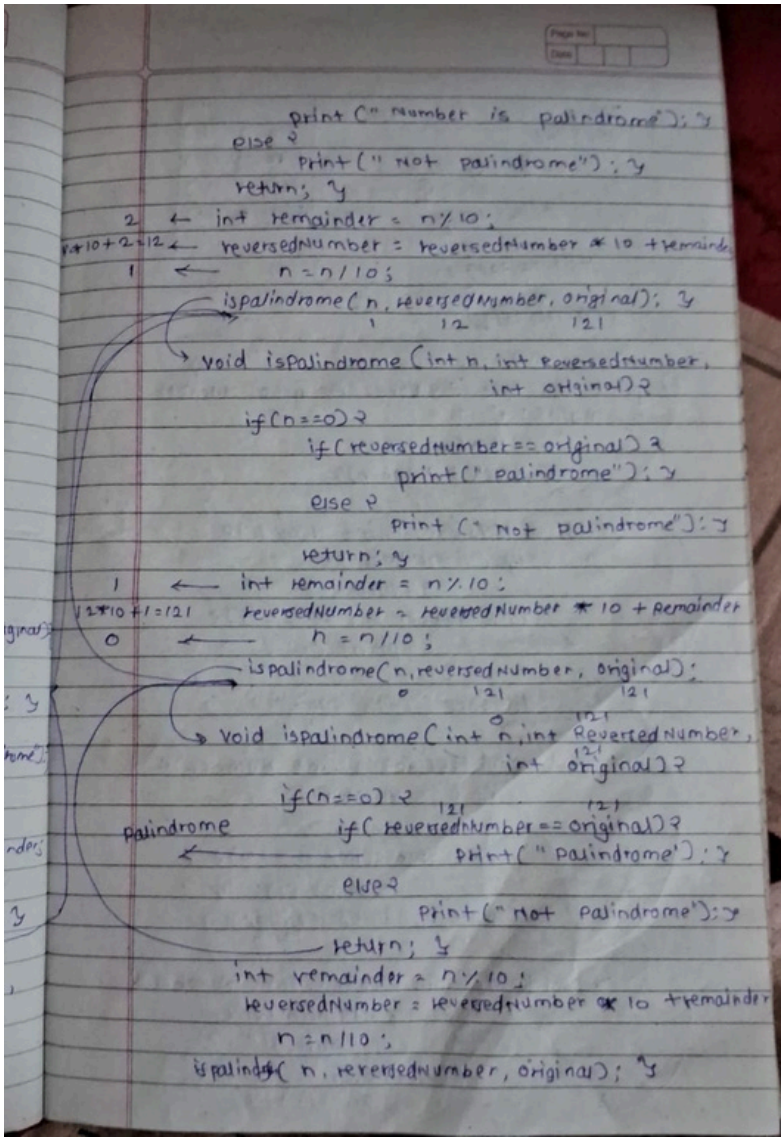
12 ← n = n/10;

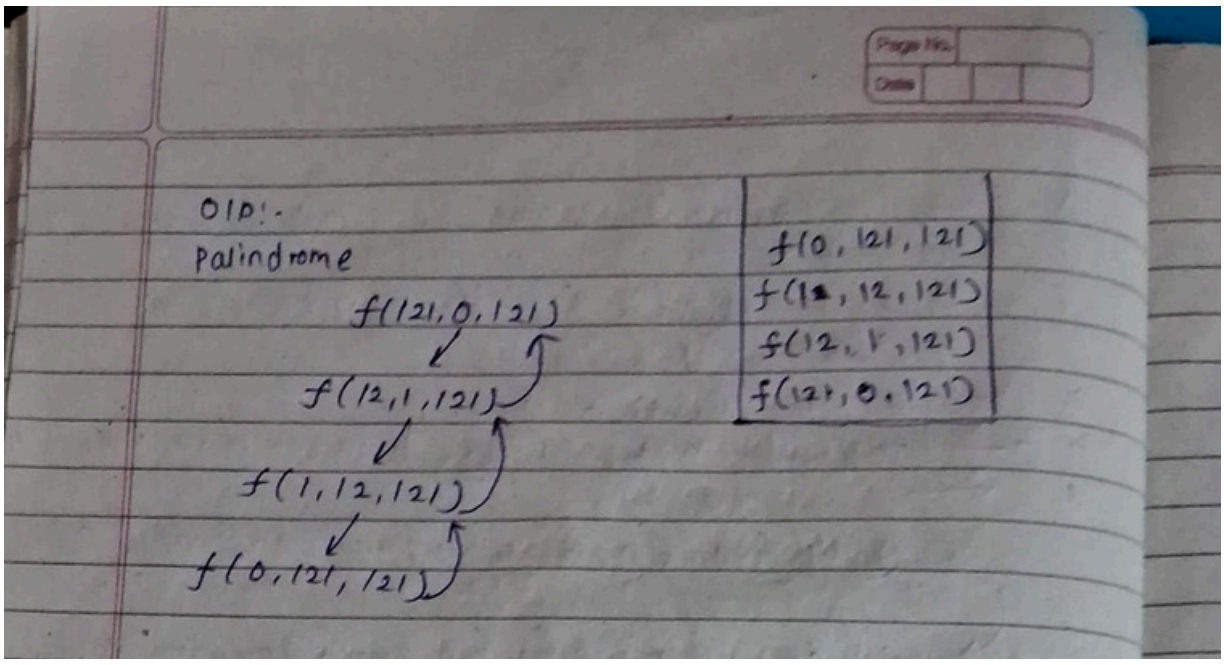
ispalindrome(12, 1, 121);

void ispalindrome(int n, int reversedNumber, int original)

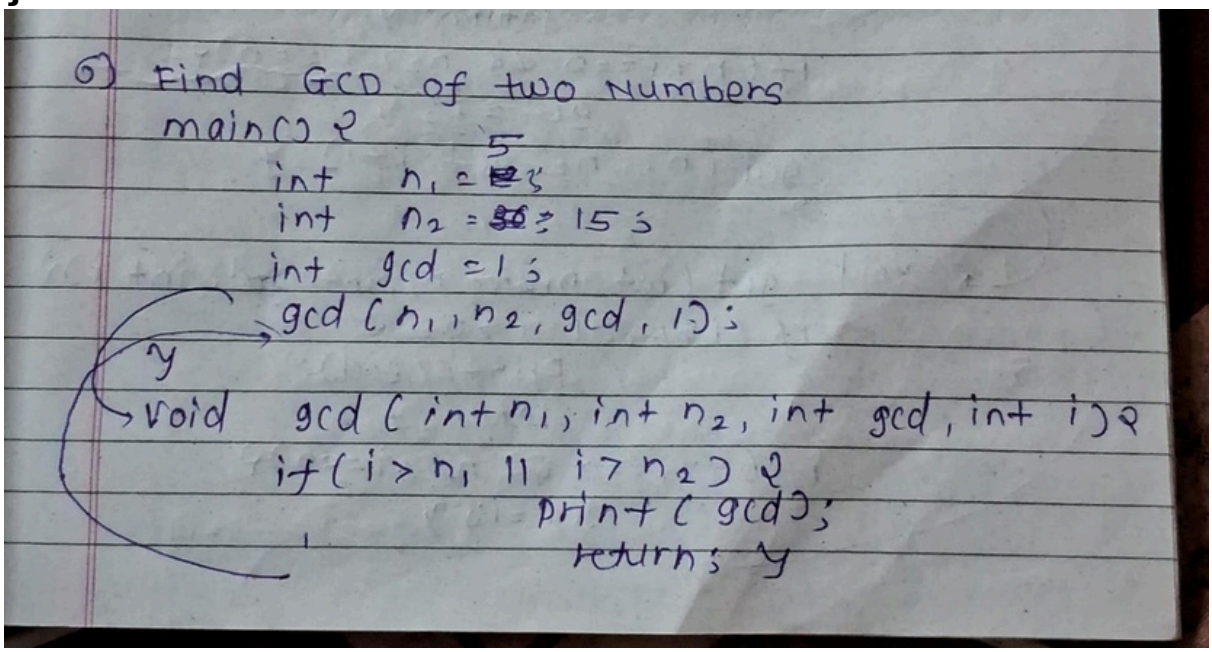
if(n == 0)

if(original == n)


print (" number is palindrome"); }
else {
print (" not palindrome"); }
return; }
int remainder = n%10;
reversedNumber = reversedNumber * 10 + remainder;
n = n/10;
ispalindrome(n, reversedNumber, original);
void ispalindrome(int n, int reversedNumber, int original){
if(n==0){
if(reversedNumber == original)
print(" palindrome"); }
else {
print (" not palindrome"); }
return; }
int remainder = n%10;
reversedNumber = reversedNumber * 10 + Remainder
n = n/10;
ispalindrome(n, reversedNumber, original);
void ispalindrome(int n, int ReversedNumber, int original){
if(n==0){
if(reversedNumber == original){
print(" palindrome"); }
else {
print (" not palindrome"); }
return; }
int remainder = n%10;
reversedNumber = reversedNumber * 10 + remainder
n = n/10;
ispalindrome(n, reversedNumber, original); }

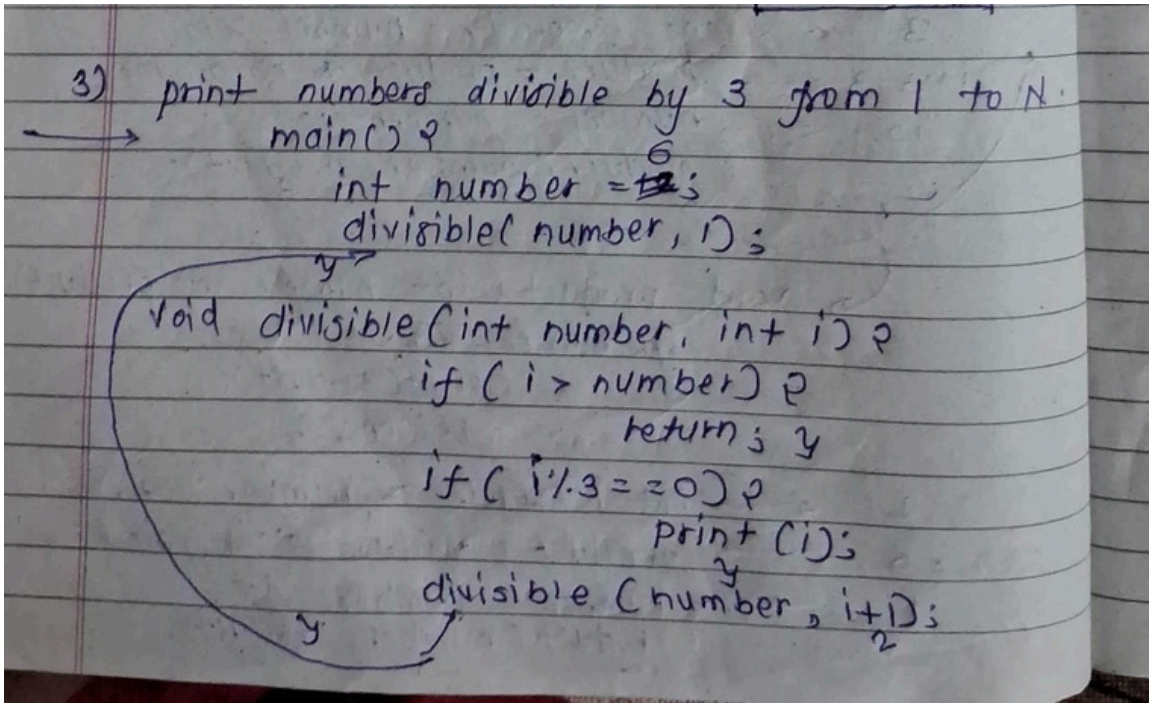

OIP:-
Palindrome
f(121,0,121)
f(12,1,121)
f(1,12,121)
f(0,121,121)
f(0, 121, 121)
f(12, 12, 121)
f(12, 1, 121)
f(121, 0, 121)

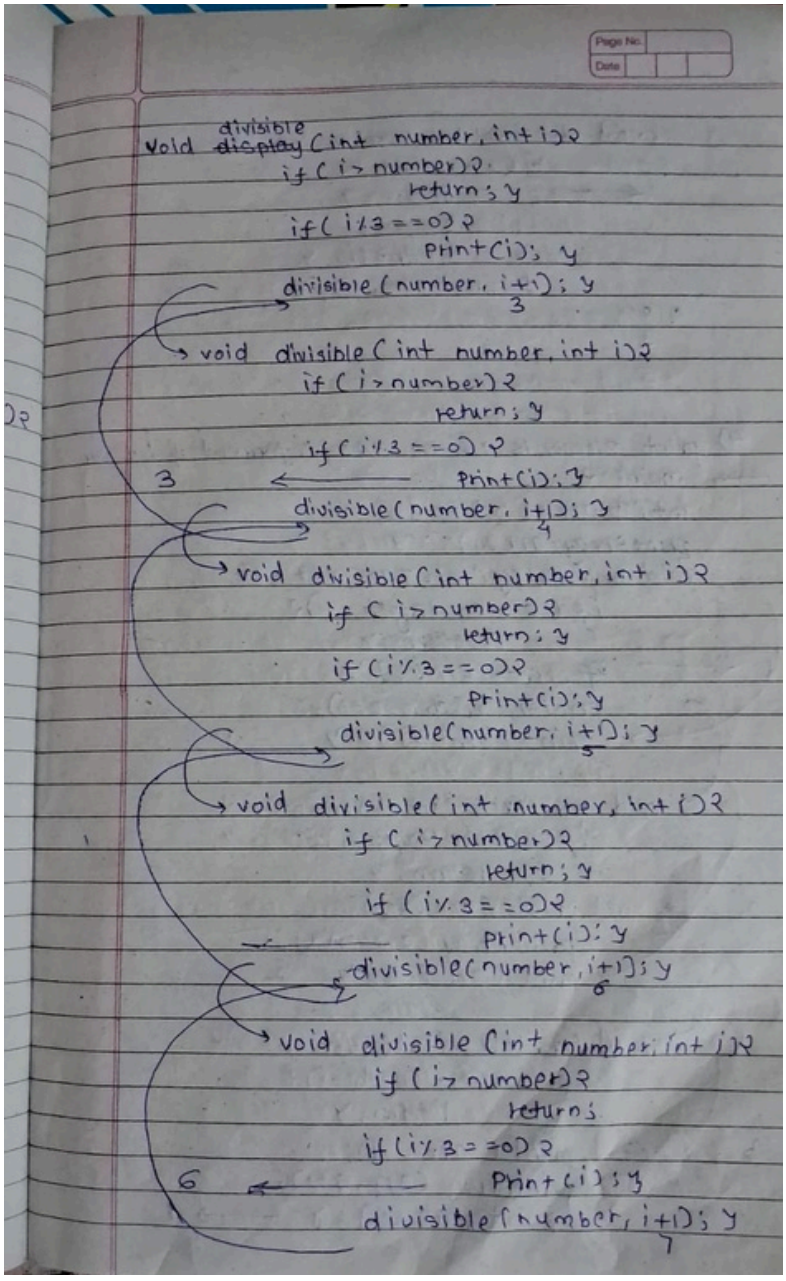
```
13. class gcd{
        public static void main(String[]args){
                int n1=12; int n2=36;
                int                gcd=1;
                gcdTwo(n1,n1,1,gcd);

}

        public static void gcdTwo(int n1,int n2,int i,int gcd){
                if(i>n1||i>n2){
                        System.out.println(gcd);
                        return;}
                if(n1%i==0 && n2%i==0){
                        gcd=i;}
                gcdTwo(n1,n2,i+1,gcd);

}


}
```

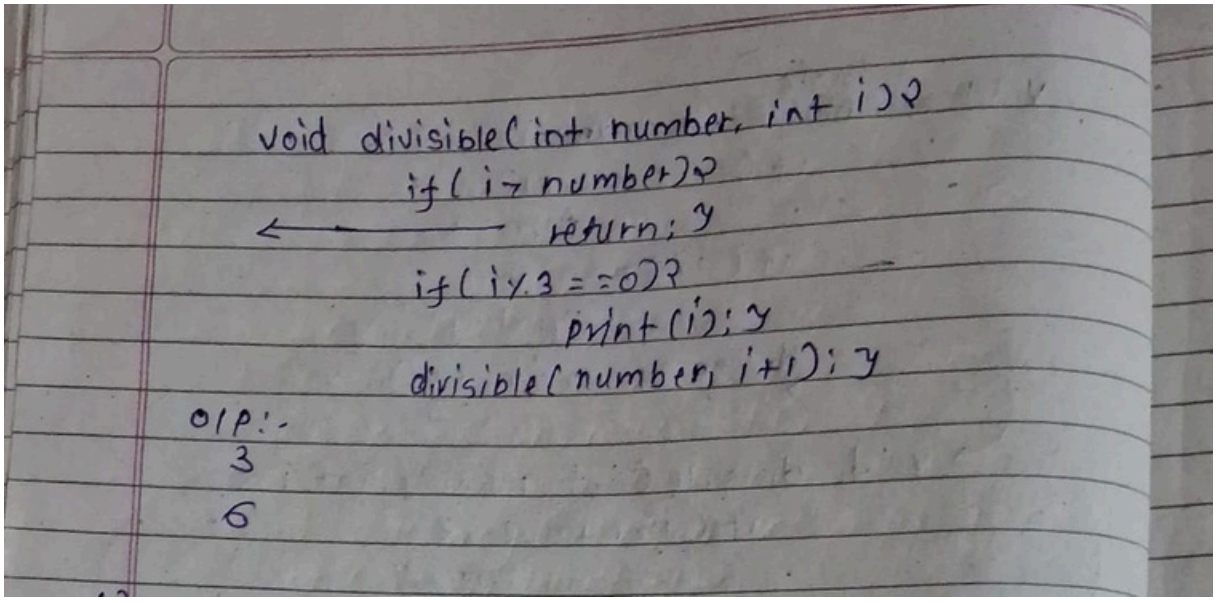


```
if (n1%i==0 && n2%i==0)?
        gcd = i; y
gcd(n1, n2, gcd, i+1); y

void gcd(int n1, int n2, int gcd, int i)?
  if (i > n1 || i > n2)?
        print(gcd);
        return; y
  if (n1%i==0 && n2%i==0)?
        gcd = i; y
  gcd(n1, n2, gcd, i+1); y

void gcd(int n1, int n2, int gcd, int i)?
  if (i > n1 || i > n2)?
        print(gcd);
        return; y
  if (n1%i==0 && n2%i==0)?
        gcd = i; y
  gcd(n1, n2, gcd, i+1); y

void gcd(int n1, int n2, int gcd, int i)?
  if (i > n1 || i > n2)?
        print(gcd);
        return; y
  if (n%i==0 && n2%i==0)?
        gcd = i; y
  gcd(n1, n2, gcd, i+1); y

void gcd(int n1, int n2, int gcd; int i)?
  if (i > n1 || i > n2)?
        Print(gcd);
        return; y
  if (n%i==0 && n2%i==0)?
        gcd = i; y
```

```
14. class printDivisible{
        public static void main(String[]args){
            int number=12;
            printAllDivisible(number,1);
    }
        static void printAllDivisible(int number,int i){
            if(i>number){
                return;}
            if(i%3==0){
                System.out.println(i);
                }
            printAllDivisible(number,i+1);
}
}
```



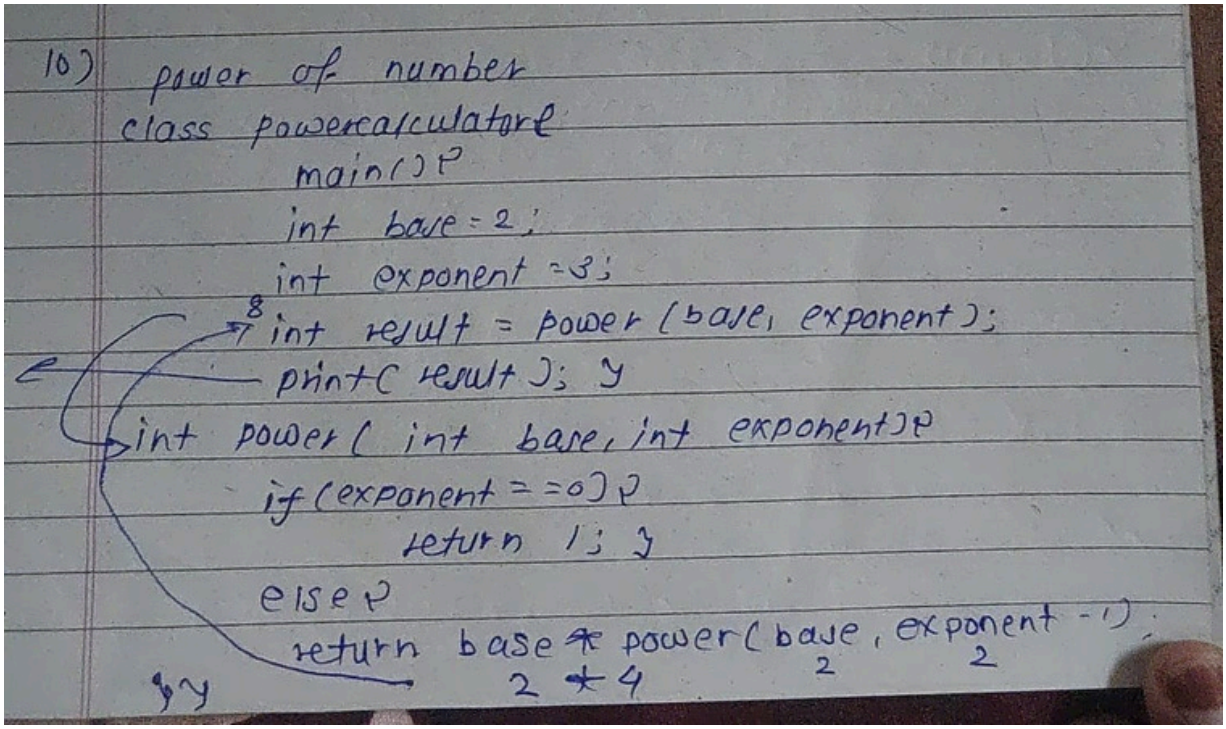

void divisible(int number, int i){
if(i>number){
return;
if(i%3==0){
Print(i);
divisible(number, i+i);
3
void divisible(int number, int i){
if(i>number){
return;
if(i%3==0){
3
Print(i);
divisible(number, i+1);
4
void divisible(int number, int i){
if(i>number){
return;
if(i%3==0){
Print(i);
divisible(number, i+1);
5
void divisible(int number, int i){
if(i>number){
return;
if(i%3==0){
Print(i);
divisible(number, i+1);
6
void divisible(int number, int i){
if(i>number){
return;
if(i%3==0){
6
Print(i);
divisible(number, i+1);
7



15. public class PowerCalculator {

```
    public static int power(int base, int exponent)
{
        if (exponent == 0) {
            return 1;
        } else {
            return base * power(base, exponent - 1);
        }
    }

    public static void main(String[] args) {
        int base = 2;
        int exponent = 3;
```

```
        int result = power(base, exponent);
        System.out.println(base + "^" + exponent +
" = " + result);
    }
}
```



```
int power(int base, int exponent){
    if(exponent == 0){
        return 1;}
    else{
        return base * power(base, exponent-1);
    }
}
```

2 * 2 = 4

```
int power(int base, int exponent){
    if(exponent == 0){
        return 1;}
    else{
        return base * power(base, exponent-1);
    }
}
```

2 * 1

```
int power(int base, int exponent){
    if(exponent == 0){
        return 1;}
    else{
        return base * power(base, exponent-1);
    }
}
```