Main.java

Share

```java
//   Print elements of an array (forward and backward)

class Main {
    public static void revFor(int [] arr,int i){
        if(i>=arr.length){
            System.out.println(" ");
            return;
        }
        System.out.print(arr[i]);
        revFor(arr,i+1);
        System.out.print(arr[i]);
    }
    public static void main(String[] args) {
        int []arr =  {1,2,3,4};
        revFor(arr,0);
    }
}
```

Output

```
1234
4321
=== Code Execution Successful ===
```

```java
//  Find the sum of digits of a number

class Main {
    public static int sum_digits(int num ){
        if(num<=0){
            return 0;
        }
        int rem = num % 10;
        return(rem + sum_digits(num/10));
    }
    public static void main(String[] args) {
        System.out.print(sum_digits(1234));
    }
}
```

Output

```
10
=== Code Execution Successful ===
```

```java
//  Find the product of digits of a number

class Main {
    public static int prod_digits(int num){
        if(num<=0){
            return 1;
        }
        int rem = num % 10;
        return(rem * prod_digits(num/10));
    }
    public static void main(String[] args) {
        System.out.print(prod_digits(231));
    }
}
```

Output

```
6
=== Code Execution Successful ===
```

```java
//  Count number of digits in a number

class Main {
    public static int digits(int num,int count){
        if(num<=0){
            return count;
        }
        return digits(num / 10, count + 1);
    }
    public static void main(String[] args) {
        System.out.print(digits(98765,0));
    }
}
```

Output

```
5
=== Code Execution Successful ===
```

```java
//  Find the maximum element in an array

class Main {
    public static int max_ele(int[] arr,int i,int max){
        if(i>=arr.length){
            return max;
        }
        if(max<arr[i]){
            max = arr[i];
        }
        return max_ele(arr,i+1,max);
    }
    public static void main(String[] args) {
        int [] arr = {2,5,9,1,6};
        System.out.print(max_ele(arr,0,0));
    }
}
```

Output

```
9
=== Code Execution Successful ===
```

```java
//  Check if an array is sorted (strictly increasing)

class Main {
    public static boolean isSorted(int[] arr,int i){
        if(i>=arr.length-1){
            return true;
        }
        if(arr[i+1]<=arr[i]){
            return false;
        }
        return isSorted(arr,i+1);
    }
    public static void main(String[] args) {
        int [] arr1 = {1,2,3,4};
        System.out.println(isSorted(arr1,0));
        int [] arr2 = {1,2,2,3};
        System.out.println(isSorted(arr2,0));
    }
}
```

Output

```
true
false

=== Code Execution Successful ===
```

```java
//   Check if a number is prime

class Main {
    public static boolean isPrime(int num,int i){
        if(i>Math.sqrt(num)){
            return true;
        }
        if(num%i==0 && num!=2){
            return false;
        }
        return isPrime(num,i+1);
    }
    public static void main(String[] args) {
        System.out.println(isPrime(2,2));
        System.out.println(isPrime(10,2));
    }
}
```

Output

```
true
false

=== Code Execution Successful ===
```

```java
//  Find the first index of an element in an array

class Main {
    public static int first_index(int []arr,int key, int i){
        if(i>arr.length){
            return -1;
        }
        if(arr[i]==key){
            return i;
        }
        return first_index(arr,key,i+1);
    }
    public static void main(String[] args) {
        int []arr ={4, 2, 7, 7, 9};
        System.out.println(first_index(arr,7,0));
    }
}
```

Output

```
2

=== Code Execution Successful ===
```

```java
//  Find the last index of an element in an array

class Main {
    public static int last_index(int []arr,int key, int i){
        if(i<0){
            return -1;
        }
        if(arr[i]==key){
            return i;
        }
        return last_index(arr,key,i-1);
    }
    public static void main(String[] args) {
        int []arr ={4, 2, 7, 7, 9};
        System.out.println(last_index(arr,7,arr.length-1));
    }
}
```

Output

```
3

=== Code Execution Successful ===
```

Main.java

Share Run

```java
//  Reverse a number using recursion

class Main {

    public static int reverse(int num ,int revNum){
        if(num<=0){
            return revNum;
        }
        int rem = 0;
        rem = num%10;
        return reverse(num/10,revNum*10+rem);
    }
    public static void main(String[] args) {
        System.out.println(reverse(1234,0));
    }
}
```

Output

```
4321

=== Code Execution Successful ===
```

```java
//  Reverse a number using recursion

class Main {

    public static int counting(int num ,int D,int count){
        if(num<=0){
            return count;
        }
        int rem = 0;
        rem = num%10;
        if(rem == D){
            count ++;
        }
        return counting(num/10,D,count);
    }
    public static void main(String[] args) {
        System.out.println(counting(717237,7,0));
    }
}
```

Output

```
3

=== Code Execution Successful ===
```

```java

class Main {

    public static boolean isPalindrome(int num ,int OrgNum ,int revNum){
        if(num<=0){
            return revNum==OrgNum;
        }
        int rem = 0;
        rem = num%10;
        return isPalindrome(num/10,OrgNum,revNum*10 + rem);
    }
    public static void main(String[] args) {
        System.out.println(isPalindrome(123,123,0));
    }
}
```

Output

```
false

=== Code Execution Successful ===
```

```java
// Find GCD (HCF) of two numbers using recursion

class main{
    static int gcd(int num1,int num2){
        if(num1 == 0 || num2 == 0){
            return Math.max(num1,num2);
        }
        return gcd(Math.max(num1,num2) % Math.min(num1,num2),Math.min
            (num1,num2));
    }
    public static void main(String[] args) {
        System.out.println(gcd(24,36));
    }
}
```

Output

```
12

=== Code Execution Successful ===
```

```java
//  Print all numbers from 1 to N divisible by 3

class main{
    public static void dividebyThree(int N,int i){
        if(i>N){
            return;
        }
        if(i%3==0){
            System.out.println(i);
        }
        dividebyThree(N,i+1);

    }
    public static void main(String[] args) {
       dividebyThree(10,1);
    }
}
```

Output

```
3
6
9

=== Code Execution Successful ===
```

```java
//  Find power of a number using recursion

class main{
    public static int powerFinder(int A,int B){
        if(B==0){
            return 1;
        }
        return A*powerFinder(A,B-1);
    }
    public static void main(String[] args) {
       System.out.println(powerFinder(2,4));
    }
}
```

Output

```
16

=== Code Execution Successful ===
```