Code : 1

```
package Topic_01_GettingStartedBasics;

import java.util.Scanner;
public class A_IsANumberPrime_1 {
	public static void main(String[] args) {
		Scanner scn = new Scanner(System.in);
		int t = scn.nextInt();
		for (int i = 0; i < t; i++) {
			CheckPrimeWithDisplay(scn.nextInt());
		}
	}

	static void CheckPrimeWithDisplay(int number) {
		int flag = 0;
		for (int i = 2; i < number; i++) {
			if (number % i == 0) {
				flag = 1;
				break;
			}
		}
		if (flag == 0) {
			System.out.println("prime");
		} else {
			System.out.println("not prime");
		}
	}
}
```

Code : 2

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class B_PrintAllPrimesTillN_2  {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int low = s.nextInt();
		int high = s.nextInt();
		PrintPrimeNumbersBetweenRange(low, high);
	}

	static void PrintPrimeNumbersBetweenRange(int low, int high) {
		for (int i = low; i <= high; i++) {
			int flag = CheckPrime(i);
			if (flag == 0)
				System.out.println(i);
		}
	}

	static int CheckPrime(int number) {
		int isPrime = 0;
		for (int i = 2; i*i <= number; i++) {
			if (number % i == 0) {
				isPrime = 1;
				break;
			}
		}
		return isPrime;
	}
}
```

Code : 3

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class C_PrintAllFibTillN_3  {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		PrintFibonacciNumbersUptoValue_N(s.nextInt());
	}

	static void PrintFibonacciNumbersUptoValue_N(int n) {
		int a = 0;
		int b = 1;
		int c = 0;
		for (int i = 0; i < n; i++) {
			System.out.println(a);
			c = a + b;
			a = b;
			b = c;
		}
	}
}
```

Code : 4

```
package Topic_01_GettingStartedBasics;

import java.util.Scanner;
public class D_CountDigitsInANumber_4  {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int n =s.nextInt();
		CountDigitsInANumber(n);
	}

	static void CountDigitsInANumber(int number) {

		int c = 0;
		while (number != 0) {
			number = number / 10;
			c++;
		}
		System.out.println(c);
	}
}
```

Code : 5

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class E_DisplayDigitOfANumber_5 {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		int n = sc.nextInt();
		int countDigits = countDigits(n);
		int div = (int) Math.pow(10, countDigits - 1);
		while (div != 0) {
			int q = n / div;
			System.out.println(q);
			n = n % div;
			div = div / 10;
		}
	}

	static int countDigits(int n) {
		int count = 0;
		while (n != 0) {
			n = n / 10;
			count++;
		}
		return count;
	}
}
```

Code : 6

```
package Topic_01_GettingStartedBasics;

import java.util.Scanner;
public class E_InverseOfANumber {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int number = s.nextInt();
		System.out.println(inverse_of_a_number(number));
	}

	static int inverse_of_a_number(int number) {
		int count = 1;
		int rem;
		int ans = 0;
		while (number != 0) {
			rem = number % 10;
			number = number / 10;
			ans = ans + count * ((int) Math.pow(10, rem - 1));
			count++;
		}
		return ans;
	}
}
```

Code : 7

```
package Topic_01_GettingStartedBasics;

import java.util.Scanner;
public class F_ReverseOfANumber_6 {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int n = s.nextInt();
		ReverseNumber(n);
	}

	static void ReverseNumber(int n) {
		int temp = 0;
		while (n != 0) {
			temp = n % 10;
			System.out.println(temp);
			n = n / 10;
		}
	}
}
```

Code : 8

```
package Topic_01_GettingStartedBasics;

import java.util.Scanner;
public class F_RotateANumber {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int number = s.nextInt();
		int howMuchToRotate = s.nextInt();
		rotateANumber(number, howMuchToRotate);
	}

	static void rotateANumber(int n, int howMuchToRotate) {
		int nod = countDigits(n);
		int k = howMuchToRotate;
		k = k % nod;
		if (k < 0) {
			k = k + nod;
		}

		int div = (int) Math.pow(10, k);
		int mult = (int) Math.pow(10, nod - k);

		int q = n / div;
		int r = n % div;
		int ans = r * mult + q;
		System.out.println(ans);
	}

	static int countDigits(int number) {
		int c = 0;
		while (number != 0) {
			number = number / 10;
			c++;
		}
		return c;
	}
}
```

Code : 9

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class G_GCDAndLCM {
   public static void main(String[] args){
      Scanner scn = new Scanner(System.in);
      int n1 = scn.nextInt();
      int n2 = scn.nextInt();

      int temp1 = n1;
      int temp2 = n2;
      while(temp1 % temp2 != 0){
         int rem = temp1 % temp2;
         temp1 = temp2;
         temp2 = rem;
      }

      int gcd = temp2;
      int lcm = (n1 * n2) / gcd;
      System.out.println(gcd);
      System.out.println(lcm);
   }
}
```

Code : 10

```
package Topic_01_GettingStartedBasics;

import java.util.*;

public class H_PrimeFactorisationOfANumber {
   public static void main(String[] args){

      Scanner scn = new Scanner(System.in);
      int n = scn.nextInt();

      for(int div = 2; div * div <= n; div++){
         while(n % div == 0){
            System.out.print(div  + " ");
            n = n / div;
         }
      }

      if(n > 1){
         System.out.print(n);
      }

   }
}
```

Code : 11

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class I_CaseOfBenjaminBulb {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		int n = sc.nextInt();
		for (int i = 1; i * i <= n; i++) {
			System.out.println(i * i);
		}
	}
}
```

Code : 12

```
package Topic_01_GettingStartedBasics;

import java.util.*;
public class J_PythagoreanTriplet {
	public static void main(String[] args) {
		// write your code here
		Scanner scn = new Scanner(System.in);
		int a = scn.nextInt();
		int b = scn.nextInt();
		int c = scn.nextInt();

		int max = a;
		max = b > max ? b : a;
		max = c > max ? c : a;

		if (max == a) {
			if ((a * a) == (b * b) + (c * c)) {
				System.out.println(true);
			} else {
				System.out.println(false);
			}
		} else if (max == b) {
			if ((b * b) == (a * a) + (c * c)) {
				System.out.println(true);
			} else {
				System.out.println(false);
			}
		} else {
			if ((c * c) == (b * b) + (a * a)) {
				System.out.println(true);
			} else {
				System.out.println(false);
			}
		}

	}
}
```