

while ($n \% 2 == 0$)

$n = n / 2;$

if ($n == 1$)

System.out.println ("Yes");

else

System.out.println ("No");

Day-5

Java: Practicing loops

for (initialization; condition; update)

without initialization

↓

int i = 0

for (; i <= 10; i += 2)

System.out.println (i);

y

without condition

int i = 0

for (; ; i += 2)

if (i > 10)

break;

System.out.println (i);

y

without update

int i = 0;

for (; ;)

if (i > 10)

break;

System.out.println (i);

y i += 2;

Q) Given a value of N, print only those numbers from 1 to N
that have sum of digits = k

1

11

20

2

12

21

3

13

22

4

14

23

5

15

24

6

16

25

7

17

26

8

18 → 1 + 8 = 9

27

9

19

28
29

```

import java.util.Scanner;
class Main {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        int R = sc.nextInt();
        int K = sc.nextInt();
        // Iterating over all values
        for (int N=1; N <= R; N++) {
            int sum=0;
            for (int num=N; num>0; num/=10) {
                int lastDigit = num%10;
                sum+=lastDigit;
            }
            if (sum==K)
                System.out.println(N+":"+sum);
        }
    }
}

```

while loop

```

for (int N=1, N<=R; N++) {
    int sum=0 int i=0
    while (i < N) {
        int lastDigit = i%10;
        sum+=lastDigit;
        i++;
    }
    if (sum==K)
        System.out.println(N+":"+sum);
}

```

while loop

```

while (condition) {
    // body
    // updation
}

```

```

int i=1;
while (i<=10) {
    System.out.println(i);
    i++;
}

```

No-while loop

```

do {
    // Code
}
while (condition)

```

Q) Print first N numbers that have their sum of digits = K
 Eg: sum of digit $K = 15$ N=5

$$\begin{aligned}
 & 1 \\
 & 2 \\
 & 3 \\
 & 4 \\
 & 5 \\
 & 69 \rightarrow 6+9=15 \quad \textcircled{1} \\
 & 78 \rightarrow 7+8=15 \quad \textcircled{2} \\
 & 87 \rightarrow 15 \quad \textcircled{3} \\
 & 96 \rightarrow 15 \quad \textcircled{4} \\
 & 159 \rightarrow 15 \quad \textcircled{5}
 \end{aligned}$$

Q) Print first 5 no's whose sum=15

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();           N → target count
        int K = sc.nextInt();           K → target sum
        int i = 1, count = 0;           i → cur num
        while (count < N) {
            int cur_sum = 0;
            for (int num = i, num > 0; num /= 10) {
                cur_sum += (num % 10);
                if (cur_sum == K) {
                    System.out.println(cur_sum);
                    count++;
                }
            }
            i++;
        }
        sc.close();
    }
}

```

Q) Given a positive no, check if prime or not
 prime → divisible by 1 and itself

Eg: 10 → I/P
 ↓
 multiples → 2, 5, 10
 So, not a prime

Program →

```

import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
    }
}

```

Steps → 1) take n
 2) iterate i from 2 to $n-1$
 3) If $(n \% i == 0)$
 ↓
 not prime ✓ 2 → 1, 2
 else ✓ 3 → 1, 3
 prime ✗ 4 → 1, 2, 4
 ✓ 5 → 1, 5
 ✗ 6 → 1, 2, 3, 6

```

        boolean isPrime = true;
        for (int i = 2; i <= N - 1; i++) {
            if (N % i == 0) {
                isPrime = false;
                break;
            }
        }
        System.out.println(isPrime);
    }
}

```

Day-6

→ Java loop extra questions

a) Print all prime numbers from 1 to N

Eg: $N = 25$

$2 \rightarrow 1, 2 \checkmark$
 $3 \rightarrow 1, 3 \checkmark$
 $4 \rightarrow 1, 2, 4 \times$
 $5 \rightarrow 1, 5 \checkmark$
 $6 \rightarrow 1, 2, 3, 6 \times$
 $7 \rightarrow 1, 7 \checkmark$
 $8 \rightarrow 1, 2, 4, 8 \times$
 $9 \rightarrow 1, 3, 9 \times$
 $10 \rightarrow 1, 2, 5, 10 \times$
 $11 \rightarrow 1, 11 \checkmark$

$12 \rightarrow 1, 2, 6, 12 \times$
 $13 \rightarrow 1, 13 \checkmark$
 $14 \rightarrow 1, 2, 7, 14 \times$
 $15 \rightarrow 1, 3, 5, 15 \times$
 $16 \rightarrow 1, 2, 4, 8, 16 \times$
 $17 \rightarrow 1, 17 \checkmark$
 $18 \rightarrow 1, 2, 9, 3, 18 \times$
 $19 \rightarrow 1, 19 \checkmark$
 $20 \rightarrow 1, 2, 5, 10, 20 \times$
 $21 \rightarrow 1, 3, 7, 21 \times$

$22 \rightarrow 1, 2, 11, 22 \times$
 $23 \rightarrow 1, 23 \checkmark$
 $24 \rightarrow 1, 2, 12, 24, 6, 4 \times$
 $25 \rightarrow 1, 5, 25 \times$

Prime numbers

1, 2, 3, 5, 7, 11, 13, 17, 19, 23

- ① Iterate from 2 to n → we have not iterated over 1
 ② Check if it is prime or not
 because 1 is neither prime
 not a prime

↓
 iterate from 2 to n
 if ($n \% i == 0$) → not prime
 else prime

- ③ If prime print it.

```

for (int num=2; num<=N; num++) {
    // check if num is prime
    boolean isPrime = true;
    for (int i=2; i<num; i++) {
        if (num%i==0)
            isPrime = false;
            break;
    }
    // If yes, then print it
    if (isPrime)
        System.out.println(num);
}
  
```

Q) Given a number, check if it's a power of 2 or not

$$\Leftrightarrow N=1 \Rightarrow \text{Yes } (2^0)$$

$$N=6 \Rightarrow \text{No}$$

$$N=8 \Rightarrow \text{Yes } (2^3)$$

1, 2, 4, 8, 16, 32, 64, 128, ...

Let's say the number is 2^k

$$N = \underbrace{2 * 2 * 2 * 2 * \dots * 2}_{k \text{ times}}$$

$$\frac{2^k}{2^1} \Rightarrow 2^{k-1}, 2^{k-2}, \dots, 2^3, 2^2, 2^1, 1$$

$$\begin{array}{r} 2^5 = \\ 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array} \quad \begin{array}{r} 2^9 = \\ 512 \\ 256 \\ 128 \\ 64 \\ 32 \\ 16 \\ 8 \\ 4 \\ 2 \end{array}$$

while ($n \% 2 == 0$)

$$n = n / 2;$$

if ($n == 1$)

s.o.println (true)

else

s.o.println (false)

Q) Find HCF of 2 numbers

gcd \rightarrow greatest common divisor

$$A = 36, B = 42$$

$$1, 2, 3, 6,$$

$$\downarrow$$

$$\underline{\text{HCF} = 6}$$

$$\begin{array}{r} A = 36 \\ B = 42 \end{array}$$

$$C = 36 \quad C = 42$$

$$\begin{array}{r} 36 \\ | \\ 42 \\ 36 \\ \hline 6 \end{array}$$

$$\underline{C = 6}$$

$$\textcircled{6}$$

public static void main (String[] args) {

int num1 = 36, num2 = 60, hcf = 0;

for (int i = 1; i <= num1 || i <= num2; i++) {

if (num1 % i == 0 && num2 % i == 0)

hcf = i;

s.o.println ("The HCF: " + hcf);

class Solution {

public boolean isPowerOfTwo
if (n == 0) return false;
while (n % 2 == 0)

$$n = n / 2;$$

boolean ans = (n == 1) ? true :
false;

return ans;