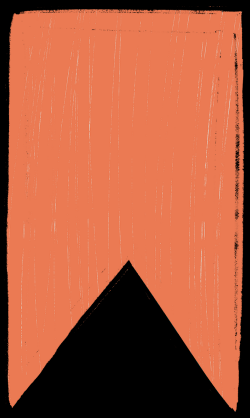
String Input

- next()
- nextLine()



hackerrank Questions

Comparison Operator

Logical Operator

- And
- OR
- Not

Scanner input with string data-type.

- next()
- nextline()

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.next();
    System.out.println(str);
```

hello

hello world

hello	world
token 1	token

```
Scanner scn = new Scanner(System.in);
String str = scn.nextLine();
System.out.println(str);
```

hello world

hello world

Hackerrank Challenges

Fahrenheit and Celsius

Problem | Submissions | Leaderboard | Discussions

You will be given Fahrenheit as input that should be stored in a double variable and print your answer in Celsius of data-type double.

Input Format

In each test case, you will get Fahrenheit as input.

nextDouble() → 0

c = (f - 32) * (5 / 9)

double c = ((f - 32) * 5) / 9;

print(c);

```
public static void main(String[] args) {
    /* Enter your code here. Read input fr
    Scanner scn = new Scanner(System.in);
    double f = scn.nextDouble();

    double c =(( f - 32 ) * 5)/9;
    System.out.println(c);
}
```

f = 47.0

15 * 5 = 75/9 = 8

8.0

Add Last Digits

Problem | Submissions | Leaderboard | Discussions

You will be given two numbers of int data-type as input, and you have to print the sum of their last digits as output.

Test Case 1:

Given Inputs: 2357 48986

Expected Output: 13

num1 → 2357 → num1 % 10 → 7 = a

num2 → 48986 → num2 % 10 → 6 = b

print(a+b);

Explanation: The last digit of 2357 is 7 and the last digit of 48986 is 6, and the sum of these last digits is 13. Hence the output is 13.

Scanner scn ⟶

int num1 = scn.nextInt()

num2

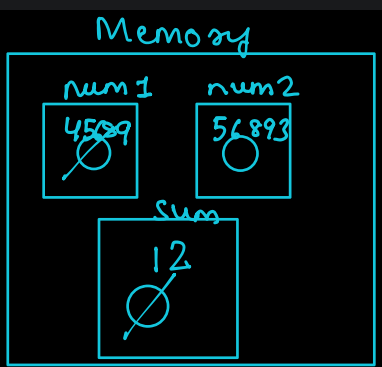
int a = num1 % 10 ; }
int b = num2 % 10 ; }

print (a+b) ;

int sum = (num1%10) + (num2%10)

```
Scanner scn = new Scanner(System.in);
int num1 = scn.nextInt();
int num2 = scn.nextInt();
int sum = (num1 % 10) + (num2 % 10);
System.out.println(sum);
```

9 3 12 12



Boolean ⟶ true / false

x = 7

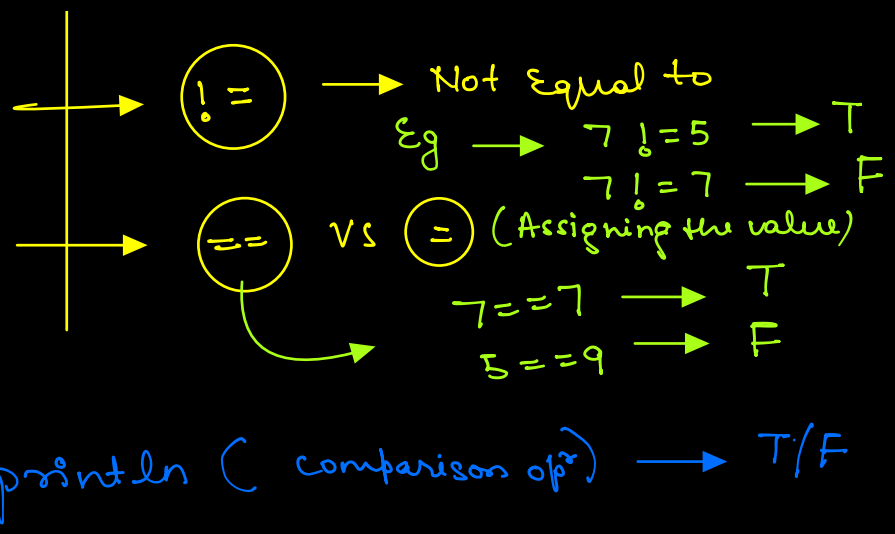
Comparison Operators

→ >, < (greater than) (less than)

eg → x > 9 → false
x < 5 → false

→ >=, <=

eg → 7 >= 7 → True
8 <= 9 → True



Greater than 100 or not

Problem | Submissions | Leaderboard | Discussions

You will be given an integer as input, you have to print true if the number is greater than 100, and false otherwise.

int x → input

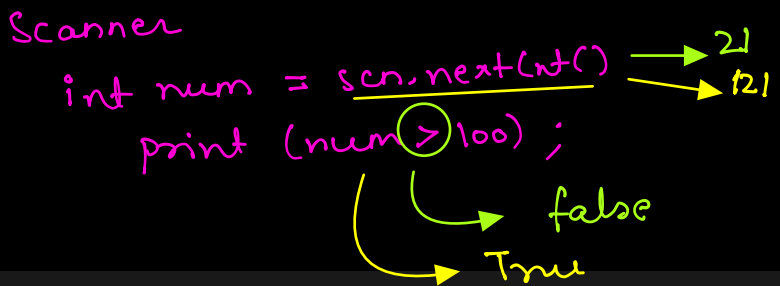
true

Test Case 1:

Input: 110

Output: true

Explanation: Since the given input is greater than 100, we printed true.




```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int num = scn.nextInt();
    System.out.println(num > 100);
}
```

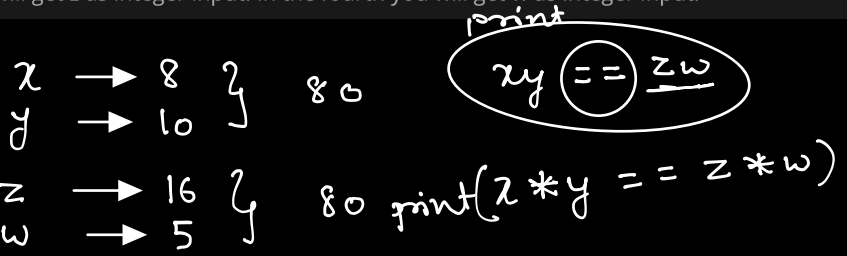
xyzw

Problem | Submissions | Leaderboard | Discussions

You will be given four integer inputs x, y, z, w. Print true if x*y is equal to z*w and false otherwise.

Input Format

For each test-case In the first you will get x as integer input. In the second you will get y as integer input. In the third you will get z as integer input. In the fourth you will get w as integer input.




```
        /* Enter your code here. Read input from STDIN.
        Scanner scn = new Scanner(System.in);
        int x = scn.nextInt(); 10
        int y = scn.nextInt(); 4
        int z = scn.nextInt(); 8
        int w = scn.nextInt(); 5

        System.out.println((x*y) == (z*w));  → true

    }
}
```

Even or not

Problem | Submissions | Leaderboard | Discussions

You have to take an integer as input and print true if it is an even number and false otherwise.

Input Format

For each test-case, you will be given an integer input

num % 2 == 0 ⟶ even

num % 2 != 0 ⟶ odd

```
lic class Solution {

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    System.out.println(n % 2 == 0);
}
```

Logical Operator

- And
- OR
- not

→ perform logical operation on value and variables

And logical (&&)

cond1 && cond2

Truth Table

cond1	cond2	Result
T	T	T
T	F	F
F	T	F
F	F	F

```
a = 10, b = 20, c = 20

condition1: a < b  → 10 < 20 → True
condition2: b == c → 10 = 20 → True

if(condition1 && condition2)
d = a+b+c  → 50

// Since both the conditions are true
d = 50.
```

```java
public static void main(String[] args) {
    int age = 24;
    int salary = 950;

    boolean result;

    result = (age >=18 && salary > 600); // yes
    System.out.println(result); // true

    result = (age >=18 && salary > 1000); // false
    System.out.println(result);

}
```

OR Logical Operator (||)

cond1 || cond2

Cond1	Cond 2	Result
T	T	T
T	F	T
F	T	T
F	F	F

```
a = 10, b = 20, c =20

condition1: a < b   → 10<20 → True ✓
condition2: b > c   → 20>20 → False

if(condition1 || condition2)
d = a+b+c

// Since one of the condition is true
d = 50.
```

```java
public static void main(String[] args) {
    int age = 24;
    int salary = 950;

    boolean result;  // declaration

    result = (age >=18 || salary > 1000);
    System.out.println(result); // true

    result = (age >=30 || salary > 1000);
    System.out.println(result); // false
```

false

t/f

Not logical Operator (!)

! condition

cond	Result
T	F
F	T

! (3 > 7)

↓

! (false) → true

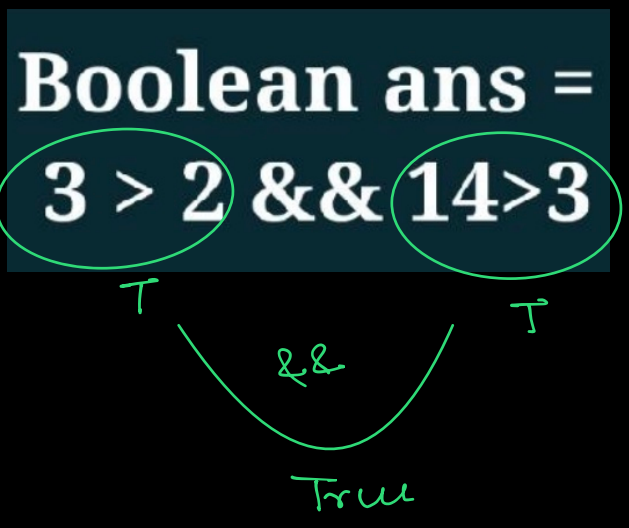
```
int age = 24;
boolean result;

result = !(age >=24); // false
System.out.println(result);

result = !(age == 25);
System.out.println(result); // true
```

---

logical Operators (Challenges)



```
Boolean ans =
40>3 && 40>50
```

True && False

False

```
Boolean ans =
40>=40 || 50>=2*25
```

cond^n

T/F

True || True

True

Short Switching

Note :

1st cond^n true and logical operator (||) do, I care about 2nd cond^n (No)

```
Boolean ans =
(2×3==4 && 6×4==9) || (4>2)
```

6 == 4

False

True

F && T/F → False

F || T → True

- Short switching
  - True with OR
  - False with AND

```
Boolean ans =
(4>5) && (3>5 && 80==2×40)
```

F

T / F

&& → False

```
Boolean ans =
(20×5==100 || 10==10) &&
(30×2==60 || 40>30)
```

T && T

True

```
Boolean ans =!(30>20)
```

→ False

True

```
Boolean ans =!(30==30)
```

!(True) → False

```
Boolean ans =
!(30>=20 || 40>=10)
```

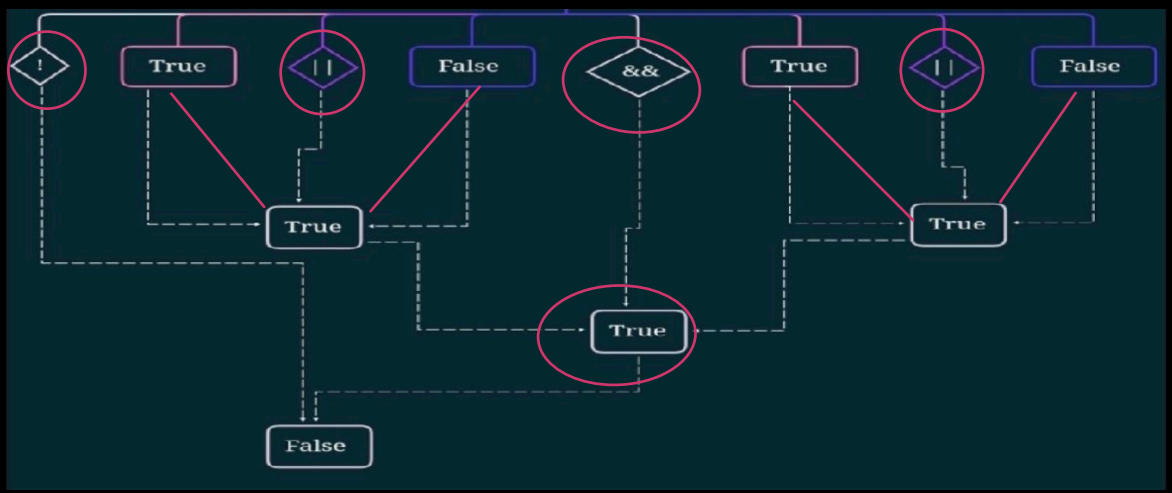
!(True) → False

```
Boolean ans =
!(20×4+40>=100 || 20==10) &&
(3×2<=60 || 4>=30)
```

!(True) → False

120

False && T/F → False



Boolean ans =
!(20%3==2)

2 == 2 true

→ False

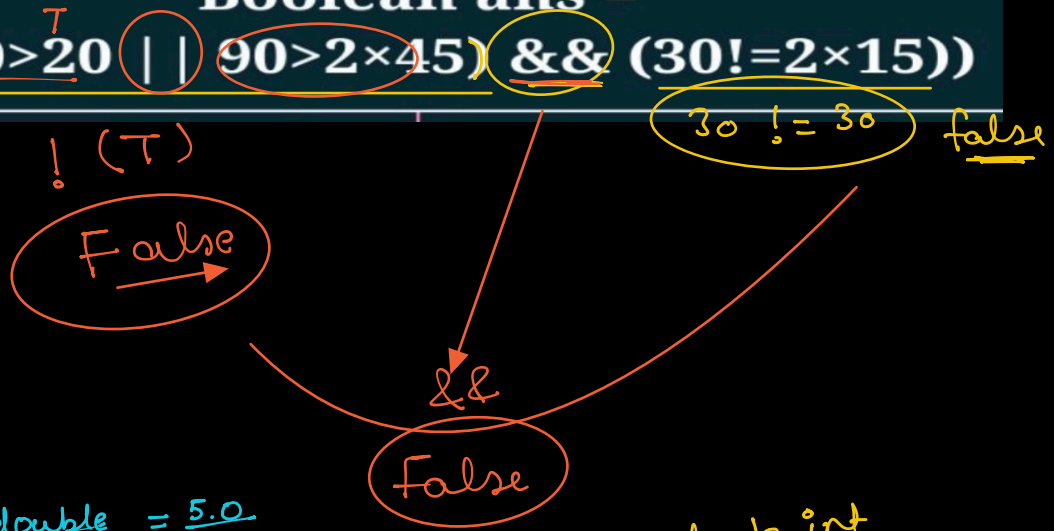
Boolean ans =
(!(40==40) && 80>36)

T T

&&

!(T) → false

Boolean ans =
(!(50>20 || 90>2×45) && (30!=2×15))



a. int x=2+3 = 5 (double = 5.0)

b. int x =3+8-29 = -18

c. int x =4 + 5.2 - 8.3 + 9.2 //1 = 18.4 - 8.3 = 10.1 = 10
conversion (double to int)
9.2 - 8.3 + 9.2

d. double x =4.1 + 8.9 + 3.5 = 16.5

e. int x= 4 * 3 / 8 + 2.5 * 2 //1 = 6
conversion
12/8 → 1 + 2.5 * 2

f. double x=22 + 4 * 2 → 22 + 8 = 30.0

g. double x=8 / 5 + 13 / 2 → 1 + 6 = 7.0
1.6 + 6 = 7.6

h. double x=8.0/5 + 13/2
8.0 / 5 → 1.6

i. double x=8.0/5 + 13.0/2 → 1.6 + 6.5 = 8.1

j. int x=392 / 10 % 10 / 2 → 39 % 10 = 9/2 = 4
(left to right)

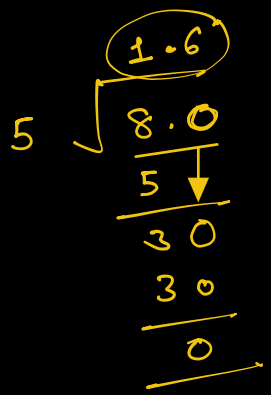
k. int x= 39%2*3 → 1 * 3 = 3

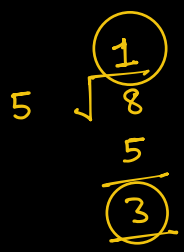
39 / 10 = 3 rem 9

% → rem.
/ → quo.

( ) > % / *
> + -

5 | 8.0 (1.6)
5↓
30
30


1.6
5
8.0
5
30
30
0


1
5
8
5
3