

# Lab: Data Types and Variables

Problems for exercises and homework for the ["Technology Fundamentals" course @ SoftUni.](#)

You can check your solutions in [Judge.](#)

## I. Integer and Real Numbers

### 1. Convert Meters to Kilometers

You will be given an integer that will be distance in meters. Write a program that converts meters to kilometers formatted to the second decimal point.

#### Examples

Input	Output
1852	1.85
798	0.80

### 2. Pounds to Dollars

Write a program that converts British pounds to US dollars formatted to 3th decimal point.

1 British Pound = 1.31 Dollars

#### Examples

Input	Output
80	104.800
39	51.090

### 3. Exact Sum of Real Numbers

Write program to enter **n** numbers and calculate and print their **exact sum** (without rounding).

#### Examples

Input	Output
3 1000000000000000000 5 10	1000000000000000015
2 0.00000000003 33333333333.3	33333333333.30000000003

#### Hints

Use **BigDecimal** to not lose precision.

## II. Data Types and Type Conversion

### 4. Town Info

You will be given 3 lines of input. On the first line you will be given the name of the town, on the second – the population and on the third the area. Use the correct data types and print the result in the following format:

"Town {town name} has population of {population} and area {area} square km".

#### Examples

Input	Output
Sofia 1286383 492	Town Sofia has population of 1286383 and area 492 square km.

### 5. Concat Names

Read two names and a delimiter. Print the names joined by the delimiter.

#### Examples

Input	Output
John Smith ->	John->Smith
Jan White <->	Jan<->White
Linda Terry =>	Linda=>Terry

### 6. Chars to String

Write a program that reads 3 lines of input. On each line you get a single character. Combine all the characters into one string and print it on the console.

#### Examples

Input	Output
a b c	abc
% 2 o	%2o
1 5	15p

p	
---	--

## 7. Reversed Chars

Write a program that takes 3 lines of characters and prints them in reversed order with a space between them.

### Examples

Input	Output
A B C	C B A
1 L &	& L 1

## 8. Lower or Upper

Write a program that prints whether a given character is upper-case or lower case.

### Examples

Input	Output
L	upper-case
f	lower-case

## 9. Centuries to Minutes

Write program to enter an integer number of **centuries** and convert it to **years, days, hours** and **minutes**.

### Examples

Input	Output
1	1 centuries = 100 years = 36524 days = 876576 hours = 52594560 minutes
5	5 centuries = 500 years = 182621 days = 4382904 hours = 262974240 minutes

### Hints

- Use appropriate data type to fit the result after each data conversion.
- Assume that a year has 365.2422 days at average ([the Tropical year](#)).

### Solution

You might help yourself with the code below:

```

Scanner scanner = new Scanner(System.in);

int centuries = Integer.parseInt(scanner.nextLine());
int years = centuries * 100;
int days = (int) (years * 365.2422);
int hours = days * 24;
int minutes = hours * 60;

System.out.printf("%d centuries = %d years = %d days = %d hours = %d minutes",
    centuries,
    years,
    days,
    hours,
    minutes
);

```

## 10. Special Numbers

A **number** is **special** when its **sum of digits** is **5, 7 or 11**.

Write a program to read an integer **n** and for all numbers in the range **1...n** to print the number and if it is special or not (**True / False**).

### Examples

Input	Output
15	1 -> False 2 -> False 3 -> False 4 -> False 5 -> True 6 -> False 7 -> True 8 -> False 9 -> False 10 -> False 11 -> False 12 -> False 13 -> False 14 -> True 15 -> False

### Hints

To calculate the sum of digits of given number **num**, you might repeat the following: sum the last digit (**num % 10**) and remove it (**sum = sum / 10**) until **num** reaches **0**.

## III. Variables

### 11. Refactor Volume of Pyramid

You are given a **working code** that finds the **volume of a pyramid**. However, you should consider that the variables exceed their optimum span and have improper naming. Also, search for variables that **have multiple purpose**.

## Code

### Sample Code

```
Scanner scanner = new Scanner(System.in);
double dul, sh, V = 0;
System.out.print("Length: ");
dul = Double.parseDouble(scanner.nextLine());
System.out.print("Width: ");
sh = Double.parseDouble(scanner.nextLine());
System.out.print("Height: ");
V = Double.parseDouble(scanner.nextLine());
V = (dul + sh + V) / 3;
System.out.printf("Pyramid Volume: %.2f", V);
```

## Hints

- **Reduce the span** of the variables by declaring them in the moment they receive a value, not before
- Rename your variables to **represent their real purpose** (example: "dul" should become length, etc.)
- Search for variables that have multiple purpose. If you find any, **introduce a new variable**.

## 12. Refactor Special Numbers

You are given a **working code** that is a solution to **Problem 9. Special Numbers**. However, the variables are **improperly named, declared before** they are needed and some of them are used for multiple things. Without using your previous solution, **modify the code** so that it is **easy to read and understand**.

## Code

### Sample Code

```
Scanner scanner = new Scanner(System.in);
int kolkko = Integer.parseInt(scanner.nextLine());
int obshto = 0;
int takova = 0;
boolean toe = false;
for (int ch = 1; ch <= kolkko; ch++) {
    takova = ch;
    while (ch > 0) {
        obshto += ch % 10;
        ch = ch / 10;
    }
    toe = (obshto == 5) || (obshto == 7) || (obshto == 11);
    System.out.printf("%d -> %b\n", takova, toe);
    obshto = 0;
    ch = takova;
}
```

## Hints

- Reduce the span of the variables by declaring them in the moment they receive a value, not before
- Rename your variables to represent their real purpose (example: "toe" should become isSpecialNum, etc.)
- Search for variables that have multiple purpose. If you find any, introduce a new variable