# More Exercise: Data Types and Variables

Problems for exercises and homework for the ["Programming Fundamentals" course @ SoftUni.](https://softuni.bg/trainings/3951/programming-fundamentals-with-java-january-2023)

You can check your solutions in [Judge.](https://judge.softuni.org/Contests/1270)

## Data Type Finder

You will receive input until you receive "**END**". Find what **data type** is the input. Possible data types are:

* Integer
* Floating point
* Characters
* Boolean
* Strings

Print the result in the following format: "**{input} is {data type} type**".

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  2.5  true  END | 5 is integer type  2.5 is floating point type  true is boolean type |
| a  asd  -5  END | a is character type  asd is string type  -5 is integer type |

## From Left to the Right

You will receive a number representing how many lines we will get as input. On the next N lines, you will receive a string with 2 numbers separated by a single space. You need to compare them. If the left number is greater than the right number, you need to print the sum of all digits in the left number, otherwise, print the sum of all digits in the right number.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  1000 2000  2000 1000 | 2  2 |
| 4  123456 2147483647  5000000 -500000  97766554 97766554  9999999999 8888888888 | 46  5  49  90 |

## **3**. Refactoring: Prime Checker

You are given a program that checks if numbers in a given range [2...N] are prime. For each number is printed "**{number} -> {true or false}**". The code, however, is not very well written. Your job is to modify it in a way that is easy to read and understand.

### Code

|  |
| --- |
| **Sample Code** |
| Scanner chetec = **new** Scanner(System.***in***);  **int** \_\_\_Do\_\_\_ = Integer.*parseInt*(chetec.nextLine()); **for** (**int** takoa = 2; takoa <= \_\_\_Do\_\_\_; takoa++) {  **boolean** takovalie = **true**;  **for** (**int** cepitel = 2; cepitel < takoa; cepitel++) {  **if** (takoa % cepitel == 0) {  takovalie = **false**;  **break**;  }  }  System.***out***.printf(**"%d -> %b%n"**, takoa, takovalie); } |

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 2 -> true  3 -> true  4 -> false  5 -> true |
| 7 | 2 -> true  3 -> true  4 -> false  5 -> true  6 -> false  7 -> true |

## 5. Decrypting Messages

You will receive a **key** (**integer**) and **n** characters afterward. Add the key to each character and append them to the **message**. In the end, print the message, which you decrypted.

### Input

* On the **first line**, you will receive the **key.**
* On the **second line**, you will receive **n** – the number of **lines** that will **follow.**
* On the next **n lines** – you will receive **lower** and **uppercase** characters from the **Latin** alphabet.

### Output

Print the **decrypted message**.

### Constraints

* The **key** will be in the interval **[0…20].**
* **n** will be in the interval **[1…20].**
* The **characters** will always be **upper** or **lower**-case letters from the **English alphabet.**
* You will receive **one** **letter** per **line.**

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| **3**  **7**  P  l  c  q  R  k  f | SoftUni |  | **1**  **7**  C  d  b  q  x  o  s | Decrypt |