

C++ Fundamentals: Exam 1

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Sunday, 15 July 2018, 09:00** (in the morning) and will close on **Sunday, 15 July 2018, 15:00**. Submit your solutions here:

<https://judge.softuni.bg/Contests/Compete/Index/1113>.

For this exam, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **int** and any floating-point input can be stored in **double**. On the Judge system, a C++ **int** is a **32-bit** signed integer and a C++ **double** is a **64-bit** IEEE754 floating point number.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 1 – Secret (Exam-1-Task-1-Secret)

We have discovered a new way that students use to cheat on exams involving single-answer questions. They send each other messages, which seem to have no correlation to the exam, but contain secret information about the correct answers to questions.

Some messages contain the numbers of questions, others contain the letters corresponding to the answers. For example, if question **15**'s correct answer is **A**, then one message will contain the number **15** encoded in it, and another will contain the letter **A** encoded in it. These values encoded in the messages we will call "**secret values**".

The encoding mechanism is simple – among the text, there will be digits. The sum of those digits is calculated, and if it corresponds to an ASCII code of a lowercase or uppercase English letter, then that letter is the **secret value**. Otherwise the sum itself is the **secret value**.

For example, the text "Hi999, 999wh9at's u2p." contains the sum $9 + 9 + 9 + 9 + 9 + 9 + 9 + 2 = 65$, which is the ASCII code of the letter 'A' – so that's the encoded letter in the message. On the other hand, the message "No5 9mu1ch." contains the sum $5 + 9 + 1 = 15$, which isn't an ASCII code for a letter, so it is just an encoded number.

You are given a single line of text (upper- and lowercase English letters, digits, punctuation, spaces, etc.) ending with a '.' (dot) character and not containing any other '.' (dot) character.

Print whatever **secret value** is encoded in the message – a letter if the value is the ASCII code of a letter, or a number otherwise.

Input

A single line on the standard input will contain the text and will contain exactly one '.' (dot) character, which will be at the end of the text.

Output

A single line, containing the **secret value**.

Restrictions

The text will contain no more than **300** characters.

The total running time of your program should be no more than **0.1s**

The total memory allowed for use by your program is **16MB**

Example I/O

Example Input	Expected Output
s8o9m9e te12xt,9he123re99,5for51secret0letterI.	I
7ext-With0ut !!,a S-ECRET L3tter.	10