C++ Fundamentals: Exam Preparation

The following tasks should be submitted to the SoftUni Judge system, which will be open starting Tuesday, 9 January 2018, 18:00 (in the afternoon) and will close on Saturday, 13 January 2018, 23:59. Submit your solutions here: https://judge.softuni.bg/Contests/Compete/Index/916.

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 4 – Visitors (JA4-Task-4-Visitors)

Some fictional website tracks visitors by assigning them an **id**, and getting their **name** and **age**. Visitors can have matching **name**s or **age**s, but will always have **unique id**s. Multiple records of a visit with the same **id** mean the user has visited the website multiple times.

The website supports the following operations for tracking and querying the tracked items:

- Adding a visit entry to the database.
  Syntax: **entry [id] [name] [age]**.
  Executed when user visits the website. **id** and **name** are strings, and **age** is a positive integer number between **1** and **99**. Note: most of these operations will be duplicates, because visitors to the website are likely to visit again, in which case the **id**, **name** and **age** values will be the same.
  Example: **entry 1A John 15**
- Querying the number of visitors with a certain **name**.
  Syntax: **name nameValue**.
  Counts the number of unique visitors which have been entered in the database with the specified **name**, when the query was given.
  Example:
  entry 1A John 15
  entry 1B Tony 16
  entry 1A John 15
  entry 1C John 86
  name John – result should be **2** (There are 2 visitors named "John" – with ids 1A and 1C)
- Querying the number of visitors within a certain **age** range.
  Syntax: **age startAge endAge**.
  Counts the number of unique visitors which have been entered in the database when the query was given, having an **age** between **startAge** (inclusive) and **endAge** (exclusive).
  Example:
  entry 1A John 15
  entry 1B Tony 16
  entry 1A John 15
  entry 1C Jebediah 87
  entry 1D Mark 16
  age 15 87 – result should be 3 (Mark, John and Tony are in the range)
- Ending the operations.
  Syntax: **end**.
  Stops the program.

Write a program which supports the operations as described above.

Input

Two or more lines containing operations as described above. The last line always contains **end**.

Output

A single line per each query command (**age** or **name**) in the input, containing a single integer number – the result of the query.

Restrictions

There will be no more than **20000** lines (operations) in the input.

Ages are between **1** (inclusive) and **100** (exclusive). Names are strings of English letters (**a**-**z**, **A**-**Z**), have a length less than **20** letters and the maximum number of unique visitor names in the input is less than **40**. Ids are strings of English letters and digits (**a**-**z**, **A**-**Z**, **0**-**9**) and tend to (not guaranteed) represent hexadecimal numbers. Ids are no more than **4** symbols long.

entry operations which have the same **id** will also have the same **name** and **age**.

The total running time of your program should be no more than **0.5s**

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

Example I/O

Example Input	Expected Output	Explanation
entry 1A John 15 entry 1B Tony 16 entry 1A John 15 entry 1C John 86 name John name Jebediah entry 1A John 15 entry 1B Tony 16 entry 1A John 15 entry 1E Jebediah 87 entry 1D Mark 16 age 15 87 end	2 0 4	When the "name John" operation is done, there are 2 unique visitors named John. When the "name Jebediah" operation is done, no visitor with the name Jebediah has been entered (yet). When the "age 15 87" operation is done, the 2 Johns (15 and 86 years old), Tony (16 years old) and Mark (16 years old) are the results.