## Sort Even Numbers

Create a program that reads one line of **integers** separated by **", "**. Then prints the **even numbers** of that sequence **sorted** in **increasing** order.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 2, 2, 4, 4, 12 | 1, 3, 5 |  | 2, 4, 6 | 2, 4, 6 |

## Sum Numbers

Create a program that reads a line of **integers** separated by **", "**. Print on two lines the **count** of numbers and their **sum.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4, 2, 1, 3, 5, 7, 1, 4, 2, 12 | 10  41 |
| 2, 4, 6 | 3  12 |

## Count Uppercase Words

Create a program that reads a line of **text** from the console. Print **all** the words that start with an **uppercase letter** in the **same order** you've received them in the text.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| The following example shows how to use Function | The  Function |
| Write a program that reads one line of text from console. Print count of words that start with Uppercase, after that print all those words in the same order like you find them in text. | Write  Print  Uppercase, |

## Add VAT

Create a program that reads one line of **double** prices separated by **", "**. Print the **prices** with **added** **VAT** for all of them. **Format** them to **2** **signs** after the decimal point. The **order** of the prices must be the **same**.  
**VAT** is equal to **20%** of the price.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 1.38, 2.56, 4.4 | 1.66  3.07  5.28 | 1, 3, 5, 7 | 1.20  3.60  6.00  8.40 |

## Filter by Age

Write a program that receives an integer **N** on the first line. On the next **N** lines, read pairs of "**[name], [age]**".Then read three lines:

* **Condition** - "**younger**" or "**older**"
* **Age** - Integer
* **Format** - "**name**", "**age**" or "**name** **age**"

Depending on the **condition**, print the correct **pairs** in the correct **format**. **Don’t use the built-in functionality from .NET. Create your methods.**

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  older  20  name age | Lucas - 20  Mia - 29  Noah - 31 | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  20  name | Tomas  Simo |  | 5  Lucas, 20 Tomas, 18 Mia, 29 Noah, 31 Simo, 16  younger  50  age | 20  18  29  31  16 |

## Action Point

Create a program that reads a collection of **strings** from the console and then **prints** them onto the **console**. Each name should be printed on a **new** **line**. Use **Action<T>**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Lucas Noah Tea | Lucas  Noah  Tea |
| Teo Lucas Harry | Teo  Lucas  Harry |
| Ashurbanipal Napoleon Caeser | Ashurbanipal  Napoleon  Caeser |

## Knights of Honor

Create a program that reads a collection of **names** as **strings** from the **console**, appends "**Sir**" in front of every name, and **prints** it back on the **console**. Use **Action<T>**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Eathan Lucas Noah Arthur | Sir Eathan  Sir Lucas  Sir Noah  Sir Arthur |
| Lucas Jade Hugo | Sir Lucas  Sir Jade  Sir Hugo |
| Ashurbanipal Napoleon Caeser | Sir Ashurbanipal  Sir Napoleon  Sir Caeser |

## Custom Min Function

Create a simple program that reads from the **console** a set of **integers** and **prints** back on the **console** the **smallest** **number** from the collection. Use **Func<T, T>**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 4 3 2 1 7 13 | 1 |
| 4 5 -2 3 -5 8 | -5 |

## Find Evens or Odds

You are given a lower and an upper bound for a range of integer numbers. Then a command specifies if you need to list all even or odd numbers in the given range. Use **Predicate<T>**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 10  odd | 1 3 5 7 9 |
| 20 30  even | 20 22 24 26 28 30 |

## Applied Arithmetics

Create a program that executes some mathematical operations on a given collection. On the **first line,** you are given **a list of numbers**. On the **next lines** you are passed **different commands** that you need to **apply to all the numbers** in the list:

* **"add"** -> add 1 to each number
* **"multiply"** -> multiply each number by 2
* **"subtract"** -> subtract 1 from each number
* **"print"** -> print the collection
* "**end**" -> ends the input

**Note: Use functions.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 5  add  add  print  end | 3 4 5 6 7 |
| 5 10  multiply  subtract  print  end | 9 19 |

## Reverse and Exclude

Create a program that reverses a collection and removes elements that are divisible by a given integer **n**. Use predicates/functions.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 5 6  2 | 5 3 1 |
| 20 10 40 30 60 50  3 | 50 40 10 20 |

## Predicate for Names

Write a program that filters a list of names according to their length. On the first line, you will be given an integer **n,** representing a name's length. On the second line, you will be given some names as strings separated by space. Write a function that prints only the names whose length is **less than or equal** to **n**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Karl Anna Kris Yahto | Karl  Anna  Kris |
| 4  Karl James George Robert Patricia | Karl |

## List of Predicates

Find all numbers in the range **1...N** that is divisible by the numbers of a given sequence. On the first line, you will be given an integer **N** – which is the end of the range. On the second line, you will be given a sequence of integers which are the **dividers**. **Use predicates/functions**.

Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  1 1 1 2 | 2 4 6 8 10 |
| 100  2 5 10 20 | 20 40 60 80 100 |