# C++ Advanced – Exam 2 (2 Dec 2018)

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++03 or the C++11 standard.

Submit your solutions here: https://judge.softuni.bg/Contests/1334/CPlusPlus-Advanced-Exam-2-2-Dec-2018

Any code files that are part of the task are provided under the folder **Skeleton**.

Please follow the exact instructions on uploading the solutions for each task.

# Task 2 – Iterator

You are given the main() function for a program that reads a sequence of numbers and calculates their sum.

To read the input, the program uses a range-based for loop to iterate an instance an Input class. On each iteration the loop accesses the next element of the sequence of numbers and adds it to the total sum.

Each item in the input is a string (a sequence of non-whitespace characters). Input items are separated by whitespaces (space, new line, tab, etc.).

The **first** string **in the input** will have the **same** value as the **last** string **in the input**. This value is called the **“end string”** and is not included in the sum calculation – it is only there to indicate when to stop reading the numbers for the sum.

The Input class has begin() and end() members so that it can be iterated by a range-based for loop. Your task is to study what type those methods return and implement it so that it can correctly function as an **iterator**, which is used to calculate the sum of the input items.

You should submit a single .zip file for this task, containing ONLY the files you created.

The Judge system has a copy of the other files and will compile them, along with your file, in the same directory.

### Hint

Remember that

for (auto item : container) { /\*body code\*/ }

is equivalent to:

for (auto iterator = container.begin(); iterator != container.end(); ++iterator) {  
 auto item = \*iterator;  
 /\*body code\*/  
}

### Restrictions

All items in the input, except the **“end string”**, will be valid integers.

There will be a small number of items in the input (less than 10).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| -- 4 8 -2 -- | 10 |
| --  4  8 -2  -- | 10 |