

## C++ Programming: Exam Variant 2 (Exam-2017-05-28)

Solutions for each task will be submitted in the form of compressed archive (.zip) files, containing .h and .cpp files.

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.

For some of the tasks in this exam you are provided with files, which the Judge system places in your submitted solution. These files are the so-called "Solution Skeleton" and, depending on the task, may require you to write specific code for your solution to work (e.g. a Solution Skeleton may contain a file with the `main()` function defined, in which case your task will usually be to implement a class or function in another file, for the program to work correctly). DO NOT attempt to edit the Solution Skeleton files – the Judge system overwrites any files from the skeleton you submit, so it won't see your changes to them. Some tasks may contain additional files you can use (and edit) if you want – if so, this will be described explicitly in the task.

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`.

## Task 1 – Transport (E2-Task-1-Transport)

Captain Grant needs your help again. He’s currently on leave, but needs to get back to his ship soon. To do that, he needs to catch a bus to the train station, and from there take a train to the naval base. But captain Grant hates waiting – he has a certain train he has to catch, but can pick from several busses, and he wants to pick a bus which arrives as close to the train departure as possible.

The transport company, which operates the busses to the station and the trains at the station, has a list of bus arrival times at the station, as well as information on the train departure time. Of course, since the company works with the military, the arrival times and the train departure time are in military time format – 4-digit numbers, the first two digits represent the hours (**00** to **23**), the next two digits represent the minutes (**00** to **59**). For example, two o’clock in the morning is **0200**, twenty minutes past four in the afternoon is **1620**, two minutes to midnight (*the time, not the Iron Maiden song*) is **2358**, etc.

Write a program which, given a list of bus arrival times and a train departure time, in military time format, finds the minimum amount of time – in minutes – between a bus arrival and the train departure (i.e. the time Grant would have to wait if he picks the “best” bus).

Note that **0** minutes is a valid result, but negative results aren’t possible.

*Hint: you can convert the military time format numbers into minutes (minutes elapsed since midnight) before calculating the time between an arrival time and the train time*

### Input

The first line of the standard input will contain a sequence of bus arrival times, in military time format, separated by single spaces.

The second line of the standard input will contain the train departure time, in military time format.

### Output

A single line containing a single non-negative integer – the minimum amount of time, in minutes, between a bus arrival and the train departure.

### Restrictions

There will be at least **1** and at most **20** bus arrival times.

The input data will be such that there will always be a valid (non-negative) minimum wait time.

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

The total memory allowed for use by your program is **5MB**

### Example I/O

Example Input	Expected Output	Explanation
0130 0004 2013 2122	69	The best bus is the one arriving at 2013 (20:13) – 69 minutes of waiting for the train
1205 1708 1241 1241	0	The train leaves at 1241 and there is a bus arriving then – 0 minutes waiting for the train