# 06. Exercise-Class-Members-and-Code-Organisation

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

Submit your solutions here: <https://judge.softuni.bg/Contests/1253/06-Exercises-on-Advanced-Class-Members-and-Code-Organization>

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 1 – Echo

You are given code for a program that manages email contacts, that has a console command-based UI. The program prints hints to the user and the user enters commands to be executed.

The program has options to disable printing the hints (though commands explicitly printing information about a contact will still print a result even if hints are disabled). That part of the code is missing. Your task is to implement the necessary functions and variables used by the existing code, so that the program compiles and accomplishes the task specified.

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.

### Restrictions & Hints

You can assume the code works correctly and the input will always lead to a valid sequence of commands. Focus on the hint-printing part, you can mostly ignore the rest (*though it would be a good idea to study the code to see how such programs may be implemented*).

In addition to the skeleton, you are given a Windows executable that behaves correctly so you can study the wanted behavior before implementing the necessary code.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| y  1  ben bendover@example.com  4  e | === Contact Manager (C) Deluxe Edition ===  Hints on? (y/n):  --- Menu ---  1. Add contact  2. Remove contact  3. Print contact  4. Toggle hints off/on  e. Exit  Please enter a choice (1, 2, 3, or Q):  Enter contact name and email:  --- Menu ---  1. Add contact  2. Remove contact  3. Print contact  4. Toggle hints off/on  e. Exit  Please enter a choice (1, 2, 3, or Q): |
| n  1  ben bendover@example.com  3  ben  4  2  ben  e | === Contact Manager (C) Deluxe Edition ===  Hints on? (y/n):  ben bendover@example.com  --- Menu ---  1. Add contact  2. Remove contact  3. Print contact  4. Toggle hints off/on  e. Exit  Please enter a choice (1, 2, 3, or Q):  Enter the name or email of a contact:  ... removed ben bendover@example.com  --- Menu ---  1. Add contact  2. Remove contact  3. Print contact  4. Toggle hints off/on  e. Exit  Please enter a choice (1, 2, 3, or Q):  Exiting...  Thank you for using Contact Manager (C) Deluxe! |

# Task 2 – Notes

You are given code for a program that reads musical note names in the solfège naming convention (Do–Re–Mi–Fa–Sol–La–Si) and translates them into NoteName objects following the English naming convention (C–D–E–F–G–A–B), however the provided code is missing the translation logic.

Your task is to study the way the provided code uses the translation logic and implement the translation logic so that the code compiles successfully and accomplishes the task described.

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(s), in the same directory.

### Restrictions & Hints

You can assume the code works correctly and the input will always lead to a valid sequence of commands. Focus on the hint-printing part, you can mostly ignore the rest (*though it would be a good idea to study the code to see how such programs may be implemented*).

In addition to the skeleton, you are given a Windows executable that behaves correctly so you can study the wanted behavior before implementing the necessary code.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Do Re Mi Fa Sol La Si unknown end | C D E F G A B ? |

# Task 3 – Vectors

You are given code for a program that reads Vectors from the console, sorts them in reverse order of their length (the Vector class defines methods for length calculation), and prints them (longest-first). For that it uses a multiset and supplies an additional template parameter that instructs the multiset how to compare two Vector objects.

Your task is to study the way the provided code and create types that allow the appropriate comparison of Vector objects, such that the resulting code accomplishes the task described.

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(s), in the same directory.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1 3  4 5  -5 10 | -5 10  4 5  1 3 |

# Task 4 – Divisible by 45

You are given a BigInt.h file with the implementation of a BigInt class which can represent positive integers of any size, can calculate sums of such integers, and has some other useful methods and operators defined. You can use this class in your program if you want to.

* The Judge system has a copy of this class and will compile your code in the same directory
* To use it, you can write #include "BigInt.h" in your code
* DO NOT submit or modify BitInt.h, as the system will overwrite it with its version. If you want to extend the functionality of that class, you will need to do it in another file

Your task is to write a program which finds all the numbers, which are divisible by 45, inside a specified range.

For this task the system only has a copy of the BigInt.h file, so the .zip file you upload should contain a file with the main() function, and you should handle input and output as described below.

### Input

Exactly 2 lines, each containing a single integer number – with an arbitrary length, but no more than 100 digits.

The first line contains the start of the range (inclusive) S

The second line contains the end of the range (exclusive) E

### Output

One or more lines, with a single integer number each, representing the numbers divisible by 45 in the given range, in ascending order (i.e. start from the smallest number divisible by 45 in the range and print each of them on a separate line).

### Restrictions

The range will be such that total numbers divisible by 45 will be no more than 100

The number of digits in the numbers specifying the range will NOT exceed 100

0 < S < E - 1

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

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  100 | 45  90 |
| 1  90 | 45 |
| 450000000000000000000000000000000000000013  450000000000000000000000000000000000000100 | 450000000000000000000000000000000000000045  450000000000000000000000000000000000000090 |

# Task 5 – Sequences

You are given code for a program that uses a Sequence class that can be iterated with a range-based for loop and can generate its elements using a templated Generator type, which overloads parameter-less operator() to generate the next element in the sequence. The provided code has two Generator types – IntegersGenerator and FibonacciGenerator, respectively generating the integer numbers starting from 0 and the Fibonacci numbers starting from 0.

The provided code reads a number from the console, generates that amount of elements in the sequence, prints the first element, then reads a number again, generates that amount of new elements in the sequence, prints the second element and so on, and continues until no more generated elements remain (meaning that the input always ends with 0).

Your task is to implement the Sequence class to support the described operations so that the code accomplishes the task described.

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(s), in the same directory.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| i  3 1 1 0 2 0 0 0 | 0 1 2 3 4 5 6 |
| f  1 5 0 0 2 0 0 0 0 | 0 1 1 2 3 5 8 13 |