

CPP MODULE 00

Exercise 00: Megaphone

Just to make sure that everybody is awake, write a program that behaves as follows:

Overview

This exercise is designed to test your understanding of basic C++ concepts and your ability to create a simple command-line program. The task is to create a program named `megaphone` that will output the input string in uppercase, unless no arguments are provided, in which case it should output a loud and unbearable feedback noise

Usage

```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS ARE ASLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
DAMNIT ! SORRY STUDENTS, I THOUGHT THIS THING WAS OFF.
$>./megaphone
* LOUD AND UNBEARABLE FEEDBACK NOISE *
```

Exercise 01: My Awesome PhoneBook

Overview

This exercise requires you to create a simple phonebook application in C++. The phonebook can store up to 8 contacts, and if a ninth contact is added, it should replace the oldest one.

Requirements

- Turn-in directory: `ex01/`
- Files to turn in: `Makefile`, `.cpp`, `{h, hpp}`
- Forbidden functions: None

Implementation Details

- Create a `PhoneBook` class with an array of `Contact` objects.
- The `PhoneBook` class should have methods to add contacts, search for contacts by index, and display all contacts.
- The `Contact` class should store information such as first name, last name, nickname, phone number, and darkest secret.

Usage

The program should accept the following commands:

- `ADD` : To add a new contact.
- `SEARCH` : To display a specific contact by index.
- `EXIT` : To quit the program.

Exercise 02: The Job Of Your Dreams

Overview

This exercise requires you to recreate a lost source file `Account.cpp` based on the provided header file `Account.hpp` and a log file.

Requirements

- Turn-in directory: `ex02/`
- Files to turn in: `Makefile` , `Account.cpp` , `Account.hpp` , `tests.cpp`
- Forbidden functions: None

Implementation Details

- Analyze the log file to understand the implementation of the `Account` class.
- Write the `Account.cpp` file to pass the tests provided in `tests.cpp` .

Usage

After writing the `Account.cpp` file, compile it and run `tests.cpp` to ensure that the output matches the expected results.

Additional Resources

- For string manipulation, refer to the [C++ string reference](#).
- For input/output manipulation, refer to the [C++ iomanip reference](#).