

HW #4

Dynamic Phone Directory

Objective: Create a C++ program that simulates a simple phone directory. The directory will manage people's names and their associated phone numbers using dynamic memory allocation for the phone numbers. This assignment will help you understand dynamic memory management, arrays of pointers, and C-style string handling.

Requirements:

- **Directory Size:** The phone directory should support a fixed number of entries (e.g., 5 entries). This number should be defined as a constant at the beginning of your program.
- **Names Array:** Use an array of constant C-style string pointers (const char*) for storing predefined names. This array should be statically allocated.
- **Phone Numbers Array:** Use an array of C-style string pointers (char*) for storing phone numbers. Memory for each phone number should be **dynamically allocated**.
- **Dynamic Memory Management:** Use new and delete to manage the memory for phone numbers. Ensure there are **no memory leaks** in your program.
- **User Interaction:** Your program should allow the user to perform the following operations:
 1. **Print the Directory:** Display all names and their associated phone numbers.
 2. **Update a Phone Number:** Allow the user to enter a name and a new phone number. The phone number for the specified name should be updated in the directory.
- **Input/Output:** Use std::cout for output and std::cin for input. Ensure that your program gracefully handles input with spaces (e.g., full names).
- **Demonstrate and test your program thoroughly** to ensure all scenarios are handled gracefully.

Sample Code Outline:Sample Code Outline:

```
#include <iostream>

const int DIRECTORY_SIZE = 5; // Example size
```

```
// Function prototypes
void printDirectory(const char* names[], char* const phoneNumbers[], int
directorySize);
void updatePhoneNumber(const char* names[], char* phoneNumbers[], int
directorySize, const char* name, const char* newNumber);
void initializeDirectory(const char* names[], char* phoneNumbers[], int
directorySize);
void cleanUp(char* phoneNumbers[], int directorySize);

int main() {
    // Initialization and user interaction code here
    return 0;
}
```

Guidelines for Implementation:

- No Standard String Library: Do not use `std::string` or functions from `<cstring>`.
- Memory Allocation: When updating a phone number, ensure you properly deallocate the old number's memory before allocating new memory for the updated number.
- Error Handling: Your program should handle cases where the user enters a name not present in the directory. It should not crash or behave unexpectedly.
- Memory Leak Checks: Before submitting, check your program for memory leaks. Ensure every new operation is matched with a delete operation.

測資:

- 1.初始化的時候:
`const char* names[DIRECTORY_SIZE] = {"Alice", "Bob", "Charlie", "David", "Eve"};`
- 2.Print the Directory
- 3.選更改電話號碼，輸入Jimmy
- 4.要印出該使用者不存在
- 5.選更改電話號碼，輸入Bob，更新Bob的電話，變成111-222-3333
- 6.再Print the Directory

請注意作業的要求，包含要使用new以及delete處理記憶體位置，以及Error handling