

```

1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 const int DIRECTORY_SIZE = 5;
6
7 char *phoneNumbers[DIRECTORY_SIZE] = {0};
8
9 const char *names[DIRECTORY_SIZE] = {"Alice", "Bob", "Charlie", "David", "Eve"};
10
11 void initializeDirectory(const char* names[], char* phoneNumbers[], int directorySize);
12 void printDirectory(const char* names[], char* const phoneNumbers[], int directorySize);
13 void updatePhoneNumber(const char* names[], char* phoneNumbers[], int directorySize, const char* name, const char* newNumber);
14 void cleanUp(char* phoneNumbers[], int directorySize);
15
16 int main() {
17
18     // 獲取當前names內Elements個數
19     int directorySize = 0;
20     for (int i=0; i<DIRECTORY_SIZE; i++){
21         if (names[i] != 0)
22             directorySize++;
23     }
24
25     // 創造放電話號碼的空間
26     for (int i=0; i<directorySize; i++){
27         phoneNumbers[i] = new char[14];
28     }
29
30     initializeDirectory(names, phoneNumbers, directorySize);
31
32     int option;
33     cout << "Update phone number(enter 0) \nLook up the directory(enter 1): ";
34     cin >> option;
35     cin.ignore();

```

```

36
37     if (!option){
38
39         char name_search[80];
40         char newNumber[14];
41
42         cout << "Whose phone number do you want to update? : ";
43         cin.getline(name_search, 80, '\n');
44
45         int find_counter = 0;
46         for (const char **name_ptr = names; *name_ptr; name_ptr++){
47
48             if (!strcmp(*name_ptr, name_search)){
49                 find_counter++;
50             }
51         }
52
53         if (!find_counter){
54             cout << "No " << name_search << " in the directory." << endl;
55             return 0;
56         }
57         else{
58             cout << "What's the new number?: ";
59             cin.getline(newNumber, 14, '\n');
60         }
61
62         updatePhoneNumber(names, phoneNumbers, directorySize, name_search, newNumber);
63         printDirectory(names, phoneNumbers, directorySize);
64
65     }
66     else if (option){
67         printDirectory(names, phoneNumbers, directorySize);
68     }
69
70     cleanUp(phoneNumbers, directorySize);
71
72     return 0;
73 }
74

```

```

75 void initializeDirectory(const char* names[], char* phoneNumbers[], int directorySize){
76
77     for (int i=0; i<directorySize; i++){
78         strcpy(phoneNumbers[i], "000-000-0000");
79     }
80
81 }
82
83 void printDirectory(const char* names[], char* const phoneNumbers[], int directorySize){
84
85     int i = 0;
86
87     cout << endl;
88     for (const char **nptr=names; *nptr; nptr++){
89         cout << *nptr << ": " << phoneNumbers[i] << endl;
90         i++;
91     }
92
93 }
94
95 void updatePhoneNumber(const char* names[], char* phoneNumbers[], int directorySize, const char* name, const char* newNumber){
96
97     for (int i=0; i<directorySize; i++){
98         if (!strcmp(names[i], name)){
99
100             delete[] phoneNumbers[i];
101             phoneNumbers[i] = new char[14];
102
103             strcpy(phoneNumbers[i], newNumber);
104             break;
105         }
106     }
107
108 }
109
110 void cleanUp(char* phoneNumbers[], int directorySize){
111
112     // deallocate memory
113     for (int i=0; i<directorySize; i++){
114         delete[] phoneNumbers[i];
115     }
116
117 }
118

```

```

Update phone number(enter 0)
Look up the directory(enter 1): 1

Alice: 000-000-0000
Bob: 000-000-0000
Charlie: 000-000-0000
David: 000-000-0000
Eve: 000-000-0000

```

```

Update phone number(enter 0)
Look up the directory(enter 1): 0
Whose phone number do you want to update? : Jimmy
No Jimmy in the directory.

```

```

Update phone number(enter 0)
Look up the directory(enter 1): 0
Whose phone number do you want to update? : Bob
What's the new number?: 111-222-3333

Alice: 000-000-0000
Bob: 111-222-3333
Charlie: 000-000-0000
David: 000-000-0000
Eve: 000-000-0000

```