CS205 C/ C++ Programming - Lab Assignment 2

Name: 邱煜（Qiu Yu）

SID: 11611127

Part1 - Analysis

Problem Statement

The problem is to design a program to recognize specific commands, do some corresponding operations and recognize invalid commands and tell users.

The specific commands are **start, stop, restart, reload, status and exit.** When the program receive commands other than **exit**, it should show some message that tell the user that it has received and recognized the command. And in this program, it displays a `>` which means it is keeping receiving commands. And after receiving the correct command, it show the message as follow:

`command <input command> recognized.`

e.g. Input **start** , it display `command start recognized.`

And if the program receives unrecognized commands, it will display `Invalid command`

e.g. Input **abc**, it display `Invalid command`

And the input is case-sensitive, so the user should input the correct form of the commands.

Besides, the codes should use `switch` statement instead of `if…else if…else…`.

Design

In my codes, I define a string array `cmds` that contains all valid commands and an integer variable `cmd_idx`. Each time the program receive a command, it search whether the received command is in `cmds`, if is, it assign the corresponding index of the command in the array(from 0 to 4) to `cmd_idx`, if not, it assign **-1** to `cmd_idx`.

Then the program use `switch` statement to show messages accordingt to `cmd_idx`.

If `cmd_idx == -1`, the program shows `Invalid command`, if `cmd_idx == 4`, which means the input command is exit, the program will invoke `exit(0)` to exit, and if `cmd_idx` is others, the program shows `command <input command> recognized.` according to the input command.

Part2 - Code

```cpp
#include <iostream>
#include <string>

#define START_CMD 0
#define STOP_CMD 1
#define RES_CMD 2
#define STATUS_CMD 3
#define EXIT_CMD 4
#define INVALID_CMD -1

using namespace std;

const string cmds[] = {"start", "stop", "restart", "status", "exit"};
const int length = sizeof(cmds)/sizeof(cmds[0]);

int searchCmd(string cmd); // search and return the input command's index in array cmds. -1 if not found.
void cmdRecog(string cmd); // print the display messages.

int main(int argc, const char * argv[]) {
    string cmd;
    while(1){
        cout << "> ";
        cin >> cmd;
        int cmd_idx = searchCmd(cmd);
        switch (cmd_idx) {
            case START_CMD:
                cmdRecog("start");
                break;
            case STOP_CMD:
                cmdRecog("stop");
                break;
            case RES_CMD:
                cmdRecog("restart");
                break;
            case STATUS_CMD:
                cmdRecog("status");
                break;
            case EXIT_CMD:
```

```
                exit(0);
                break;
            default:
                cout << "Invalid command" << endl;
                break;
        }
    }
    return 0;
}

int searchCmd(string cmd){
    int idx = INVALID_CMD;
    for (int i = 0; i < length; i++) {
        if(cmd.compare(cmds[i])==0){
            idx = i;
            break;
        }
    }
    return idx;
}

void cmdRecog(string cmd){
    cout << "command " << cmd << " recognized." << endl;
}

```

Part 3 - Result & Verification

Test case #1:

```
Input:
    start
    stop
    restart
    status
    exit
Output:
    command start recognized.
    command stop recognized.
    command restart recognized.
    command status recognized.
    Program ended with exit code: 0
```

```
> start
command start recognized.
> stop
command stop recognized.
> restart
command restart recognized.
> status
command status recognized.
> exit
Program ended with exit code: 0
```

Test case #2:

```
Input:
    safsf
    start
    Start
    safslfap
    EXIT
    exit
Output:
    Invalid command
    command start recognized.
    Invalid command
    Invalid command
    Invalid command
    Program ended with exit code: 0
```

Part 4 - Difficulties & Solutions

1. I tried to design a function to calculate a string array's length, but the size calculated inside the function is wrong, only the result calculated outside the function is correct. So I assign the `cmds` to constant variable and calculate and assign its length to a constant variable also.