CS205 C/ C++ Programming - Lab Assignment

Name: 何宜芮(Yirui He)

SID: 11811031

Part 1 - Analysis

Assignment 2 ask we to write a program to deal with several command: start, stop, restart, status, exit. When a command other than "exit" is recognized, the program will display "command <name here> recognized". It must also say "Invalid command" if the command isn't recognized. When "exit" recognized, the program should exit.

Also, commands like "start xxx" are invalid command. Commands like "   start   " is valid. And if user press enter without input anything, a new command prompt will be displayed to indicate user to enter a new command.

We only have:

```
char *commands[] = {"start", "stop", ... };

#define START_CMD      0
#define STOP_CMD       1
```

and use in the switch:

```
case START_CMD: ...
```

First, I need to get the input correctly. Then I use two functions to put them into the right form. I use a variable called "instruction" to check which instruction matches the input. Because we use "#define" to make a connection between commands and integers in advance, we are able to use switch to choose which sentence we are going to print.

Part 2 – Code

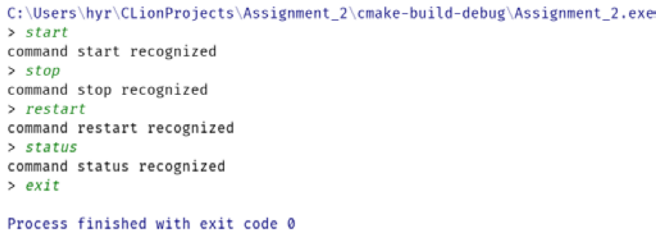
```
#include <iostream>
#include <cstring>
#include <cstdio>
#include <cstdlib>
#include <cctype>

#define START_CMD      0
#define STOP_CMD       1
#define RESTART_CMD    2
#define STATUS_CMD     3
#define EXIT_CMD       4
```

```
#define BLANK              5
#define ERROR              -1

void trim(char *, char *);

char *strlower(char *);

char *inputString(FILE *, size_t);

using namespace std;

char *command[] = {"start", "stop", "restart", "status", "exit"};

int main() {
        char *input;
        char ch;
        bool valid;
        cout << "> ";
        while (true) {
                input = inputString(stdin, 100);
                trim(input, input);
                input = strlower(input);
                int instruction = -1;
                for (int i = 0; i < 5; i++) {
                        if (strcmp(input, command[i]) == 0)instruction = i;
                }
                if (strcmp(input, "\0") == 0)instruction = 5;
                switch (instruction) {
                        case START_CMD:
                                cout << "command start recognized\n";
                                break;
                        case STOP_CMD:
                                cout << "command stop recognized\n";
                                break;
                        case RESTART_CMD:
                                cout << "command restart recognized\n";
                                break;
                        case STATUS_CMD:
                                cout << "command status recognized\n";
                                break;
                        case EXIT_CMD:
                                free(input);
                                input = NULL;
                                return 0;
```

```
            case BLANK:
                break;
            case ERROR:
                cout << "Invalid command\n";
        }
        cout << "> ";
    }
}

void trim(char *in, char *out) {
    int i, j;
    i = 0;
    j = strlen(in) - 1;
    while (isspace(in[i]) && i <= j)
        ++i;

    while (isspace(in[j]) && i <= j)
        --j;
    strncpy(out, in + i, j - i + 1);
    out[j - i + 1] = '\0';
}

char *strlower(char *str) {
    char *orign = str;
    for (; *str != '\0'; str++)
        *str = tolower(*str);
    return orign;
}


//This function is found in Stackoverflow, the webside is:
https://stackoverflow.com/questions/16870485/how-can-i-read-an-input-string-of-unknown-length
char *inputString(FILE *fp, size_t size) {
//The size is extended by the input with the value of the provisional
    char *str;
    int ch;
    size_t len = 0;
    str = (char *) realloc(NULL, sizeof(char) * size);//size is start size
    if (!str)return str;
    while (EOF != (ch = fgetc(fp)) && ch != '\n') {
        str[len++] = ch;
        if (len == size) {
            str = (char *) realloc(str, sizeof(char) * (size += 16));
            if (!str)return str;
        }
    }
    str[len++] = '\0';
    return (char *) realloc(str, sizeof(char) * len);
}
```

Part 3 - Result & Verification

Test case #1:

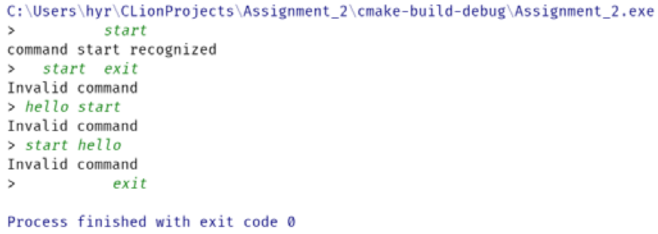


```
C:\Users\hyr\CLionProjects\Assignment_2\cmake-build-debug\Assignment_2.exe
> start
command start recognized
> stop
command stop recognized
> restart
command restart recognized
> status
command status recognized
> exit

Process finished with exit code 0
```

The output is correct.

Test case #2:

(The first input is with spaces in both sides)

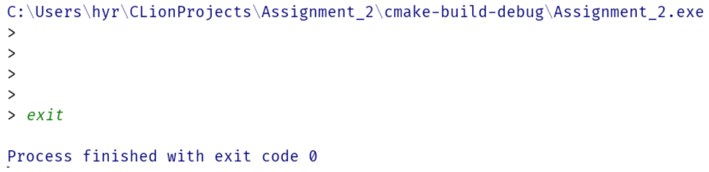


```
C:\Users\hyr\CLionProjects\Assignment_2\cmake-build-debug\Assignment_2.exe
>           start
command start recognized
>   start  exit
Invalid command
> hello start
Invalid command
> start hello
Invalid command
>           exit

Process finished with exit code 0
```

The output is correct.

Test case #3:

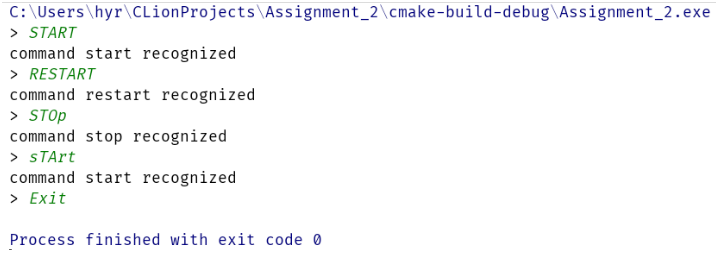
(Only press enter and only input white spaces)



```
C:\Users\hyr\CLionProjects\Assignment_2\cmake-build-debug\Assignment_2.exe
>
>
>
>
> exit

Process finished with exit code 0
```

The output is correct.

Test case #4:



```
C:\Users\hyr\CLionProjects\Assignment_2\cmake-build-debug\Assignment_2.exe
> START
command start recognized
> RESTART
command restart recognized
> STOp
command stop recognized
> sTArt
command start recognized
> Exit

Process finished with exit code 0
```

The output is correct.

Part 4 - Difficulties & Solutions

1. I want to get a string which I did not know its length in advance. So I used several methods to solve this problem. But none of them could deal with all the invalid solution safely. Finally, I asked my SA and he suggested me to search on the internet. I finally got a solution which can deal with all invalid inputs.
2. Before this assignment, I do not know how to use "#define" properly. By searching and tring, I finally get the right way to use it.