

Character Strings

Workshop 10 (out of 10 marks - 3% of your final grade)

In this workshop, you will code a C-language program that uses C style null-terminated character strings.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to code instructions that accept string input
- to code instructions that display string output
- to copy and compare character strings
- to describe to your instructor what you have learned in completing this workshop and explain the purpose of the null-terminator in a character array

SUBMISSION POLICY

The "in-lab" section is to be completed during your assigned lab section. It is to be completed and submitted by the end of the workshop period. If you attend the lab period and cannot complete the in-lab portion of the workshop during that period, ask your instructor for permission to complete the in-lab portion after the period. If you do not attend the workshop, you can submit the "in-lab" section along with your "at-home" section (with a penalty; see below). The "at-home" portion of the lab is due on the day that is two days before your next scheduled workshop (23:59).

The at-home portion of this workshop consists of some reflection questions and a brief survey.

All your work (all the files you create or modify) must contain your name, Seneca email and student number.

You are responsible to back up your work regularly.

Late submission penalties:

- In-lab portion submitted late, with at-home portion: 0 for in-lab. Maximum of 70/70 for at-home and reflection
- If any of in-lab, at-home or reflection portions is missing the mark will be zero.

IN-LAB (40%)

Download or clone workshop 10 (**WS10**) from <https://github.com/Seneca-144100/IPC-Workshops>

There are two text data-files in this lab: **starwars_directory.csv** and **jedi_master.txt**. **starwars_directory.csv** holds the phone number of all the StarWars heroes in the galaxy.

This file contains names and their phone numbers in following format:

Full Name	NPA	CO	NUMBER
Maximilian Veers,	289	555	0128

<NEWLINE>

* * * ← See delimiters note below

Full Name: length is less than or equal to 30 characters in file.

NPA: is exactly 3 characters.

CO: is exactly 3 characters.

NUMBER: is exactly 4 characters.

* Delimiters:

- ‘,’ Comma char. separates the full name and NPA fields
- ‘ ’ Space char. separates the NPA and CO fields
- ‘ ’ Space char. separates the CO and NUMBER fields

Sample Data Extract:

```
Salacious B Crumb,902 555 0167
Jar Jar Binks,867 555 0149
Biggs Darklighter,226 555 0119
Wilhelm Screamers,249 555 0122
Maximilian Veers,289 555 0128
Cornelius Evazan,343 555 0180
```

jedi_master.txt holds the names of all the Jedi Masters in the galaxy.

This file contains one full name per line that is less than or equal to 30 characters.

Open [144_w10.c](#) and complete the code of two functions to format the phone records in a string.

```
int isJediMaster(const char name[]);
```

Reads the **full-names** from the **Jedi Master file**, one by one, and **compares** each of them with the incoming **name** argument, if there is a match it will return **true (1)** otherwise **false (0)**;

*Note: Remember that you must have only **one** return statement in a function.*

This function will be used to check and see if the name read from phone directory belongs to a Jedi Master.

```
void formatJediPhoneRecord(char formattedRecord[],  
                           const char fullName[],  
                           const char npa[],  
                           const char co[],  
                           const char number[]);
```

Receives Jedi phone record through **fullname**, **npa**, **co** and **number** arguments and formats the phone record into one C-style string as follows:

```
Padme Amidala      (418) 555-0105  
Mace Windu         (438) 555-0155 Jedi Master  
Emperor Palpat..  (450) 555-0143
```

Formatting steps:

First if the name is longer than 16 characters it will shorten it to 14 and then concatenates two dots (".") to the end of it to make it exactly 16 characters. By doing this the viewer will notice if a name is shortened. *(see third line in the example above)*

Otherwise (if the name is shorter than 16) to make the name exactly 16 characters, it will concatenate a string full of space with the length of (16 – Length of name). *(see lines 1 and 2 in the example above)*

This name will be copied into **formattedRecord**.

Then concatenate the following to the **formattedRecord**:

- 1- a **space** and an **open parentheses**
- 2- **npa** string
- 3- **Close parentheses** and a **space**
- 4- **co** string
- 5- a **dash** ("-")
- 6- **number** string
- 7- if this the name is of a Jedi Master, concatenate (" **Jedi Master**") *(see second line in the example above)*

After completing the two function, the workshop should generate the following output:

```
Nien Nunb      (403) 555-0163
Baron Notluwis.. (587) 555-0155
Bib Fortuna     (780) 555-0179
Salacious B Cr.. (902) 555-0167
Jar Jar Binks   (867) 555-0149
Biggs Darkligh.. (226) 555-0119
Wilhelm Scream.. (249) 555-0122
Maximilian Veers (289) 555-0128
Cornelius Evazan (343) 555-0180
Anakin Skywalker (365) 555-0110 Jedi Master
General Grievous (416) 555-0147
Darth Maul      (437) 555-0160
Grand Moff Tar.. (519) 555-0131
Mon Mothma      (613) 555-0196
Count Dooku     (647) 555-0140
Lando Calrissian (705) 555-0132
Admiral Motti    (807) 555-0102
Wedge Antilles   (905) 555-0100
Padme Amidala    (418) 555-0105
Mace Windu       (438) 555-0155 Jedi Master
Emperor Palpat.. (450) 555-0143
Qui-Gon Jinn     (514) 555-0138 Jedi Master
Jabba the Hutt   (579) 555-0178
Admiral Ackbar   (581) 555-0120
Chewbacca        (819) 555-0168
Yoda             (873) 555-0153 Jedi Master
Boba Fett        (306) 555-0131
Luke Skywalker   (639) 555-0176
R2-D2            (867) 555-0121
C-3PO            (403) 555-0113
Han Solo         (250) 555-0161
Princess Leia    (604) 555-2121
Obi-Wan Kenobi   (365) 555-3113 Jedi Master
Darth Vader      (416) 555-4161
```

IN-LAB SUBMISSION

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload [144_w10.c](#) and [144_w10_tester.c](#) to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account (replace profname.proflastname with your professors Seneca userid):

```
~profname.proflastname/submit 144_w10_lab <ENTER>
```

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.

AT-HOME REFLECTION & SURVEY (60%)

Using the provided [reflect.txt](#) file with this workshop, provide brief answers to the following questions.

WORKSHOP QUESTIONS:

Note: when completing the workshop reflections it is a violation of academic policy to cut and paste content from the course notes or any other published source, or to copy the work of another student.

- 1- Compare a C-style string to an array of chars. What is the difference?
- 2- What format string do you use to ensure that the number of input characters read by scanf() into a C-style string does not exceed the 10 bytes of space that you have allocated for the string?
- 3- Why does strcmp(const char*, const char*) return 0 for two matching strings?
- 4- If you concatenate "Hello" to "C" how many bytes of memory do you need to store the result?

SUBJECT SURVEY:

Using the provided [survey.txt](#) file with this workshop, provide brief answers to the following questions.

- 1- What was the most interesting thing you learned this semester?
- 2- Do you feel the quizzes about the week's readings helped you learn more than you might have otherwise?
- 3- Are there any things that you particularly like about the way the course is delivered?
- 4- Are there any things that you particularly dislike about the way the course is delivered?
- 5- Is there anything you would like to see added to the way the course is delivered?
- 6- How would you rate your level of understanding of the course topics?
 - a. Very good
 - b. Pretty good
 - c. Adequate
 - d. Poor
- 7- Did you enjoy doing the workshops? Why?
- 8- The content of this course was
 - a. Too little
 - b. Just right
 - c. Too much
- 9- Did you enjoy the Lab-A activities?
- 10- Do you feel that Lab-A helped you understand how to think like a programmer?

AT-HOME SUBMISSION

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload the files [reflect.txt](#) and [survey.txt](#) to your matrix account.

Run the following script from your account (replace profname.proflastname with your professors Seneca userid):

```
~profname.proflastname/submit 144_w10_home <ENTER>
```

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.