BTP100 -- Assignment 1

Introduction

In this assignment, you are asked to examine and debug a piece of code which is near completion. You can download the code for the project from:

https://github.com/Seneca-144100/BTP-Project/tree/master/A1/MS4/

The code in the project reads the information for an entry in an address book application. All of the information is stored in structs, declared in the file contacts.h. The file contacts.c provides functions which aid in the reading of the data. The main function is in a1ms4.c and directs the operation of the other functions. Once you have downloaded the code, spend some time reading it over so that you get a good understanding of it.

Coding Part (40%)

Once you have an understanding of the code, you should run it with the following data:

(Note: due to bugs in the program, it might not act exactly like this, but this is how it SHOULD act.)

```
Please enter the contact's first name: Wilma Dee
Do you want to enter a middle initial(s)? (y or n): y
Please enter the contact's middle initial(s): N. O.
Please enter the contact's last name: Flint Rubble
Please enter the contact's street number: 30
Please enter the contact's street name: Bedrock St.
Do you want to enter an apartment number? (y or n): y
Please enter the contact's apartment number: 12
Please enter the contact's postal code: Z8Z 7R7
Please enter the contact's city: North York
Do you want to enter a cell phone number? (y or n): Y
Please enter the contact's cell phone number: 9992223333
Do you want to enter a home phone number? (y or n): Y
Please enter the contact's home phone number: 8881112222
Do you want to enter a business phone number? (y or n): Y
Please enter the contact's business phone number: 3337779999
```

After entering this data, the program should produce the following output, if it was bug-free. However, you will observe other output. You task is to find the bugs in the program and correct them so that the correct output is produced.

```
Contact Details
===============
Name Details
```

```
------------
First name: Wilma Dee
Middle initial(s): N. O.
Last name: Flint Rubble

Address Details
---------------
Street number: 30
Street name: Bedrock St.
Apartment: 12
Postal code: Z8Z 7R7
City: North York

Phone Numbers
-------------
Cell phone number: 9992223333
Home phone number: 8881112222
Business phone number: 3337779999

Structure test for Contact using functions done!
```

In order to find the bugs in the program, you will need to use the debugging techniques described in the notes and might need to use different test data to help you locate the bugs. After you have located and fixed the bugs in the program, you should complete the reflection and submit your work (the fixed source code files and reflect.txt file) as detailed below. The reflection represents the majority of the marks for the assignment and therefore should be written carefully and thoughtfully. You should expect to write 300 – 500 words to properly address the reflections. They should be written in a text document called **reflect.txt**.

Reflections (60%)

1. For each of the bugs you corrected in the code, write the original code and the corrected code and then explain what was wrong and how you corrected it.
2. The Contact struct contains other structs embedded within it. Explain how this looks when the data is laid out in memory.
3. The code in contacts.c contains some repetition of the same code several times. Can you identify the repeated code and design a function to replace it? Show the prototype for the function you designed and explain its purpose, the parameters it requires and what it will return.
4. There are several places in the code where the scanf format strings end with “%*c”. Explain what this does and why it needs to be there.
5. Most of the strings are read with the format code “%[^\n]”. Explain why this is used and why “%s” cannot be used in these situations.

Submission

You should place your name and student number on all your files. Once they are ready, transfer the files to matrix and compile and test your program using gcc with the following command:

```
gcc -Wall a1ms4.c contacts.c -o as1
```

You should then type “as1” to run your program an ensure it performs as expected and that the compilation was free from warnings. Once your program is ready, place all of your files along with reflect.txt in the same directory and execute the submit command:

```
~profFirst.profLast/submit 100a1ms4
```

Where profFirst and profLast are replaced with your professor’s first and last name.