Exploring Python tools and features - Part I

In this example, you will compile and run a program in C using the Jupyter notebook workspace provided (Buffer Overflow in C). The program is already provided as bufoverflow.c - a simple program that creates a buffer and then asks you for a name, and prints it back out to the screen.

This is the code in bufoverflow.c. You are able to download the zip file '**buffer-overflow-in-c**'. Additional instructions are provided in the **Buffer Overflow PDF**.

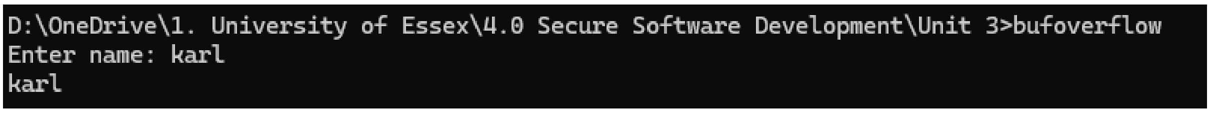
```
#include <stdio.h>

int main(int argc, char **argv)

{

char buf[8]; // buffer for eight characters

printf("enter name:");

gets(buf); // read from stdio (sensitive function!)

printf("%s\n", buf); // print out data stored in buf

return 0; // 0 as return value

{
```

Now compile and run the code. To test it, enter your first name (or at least the first 8 characters of it) you should get the output which is just your name repeated back to you.

Run the code a second time (from the command window this can be achieved by entering ./bufoverflow on the command line). This time, enter a string of 10 or more characters.

What happens?

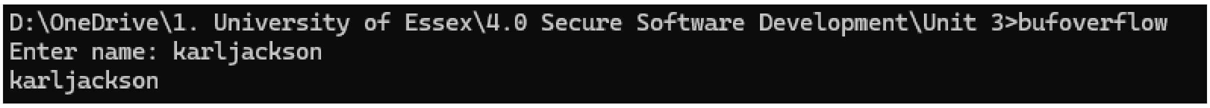
As expected, the programme returns the name as entered



```
D:\OneDrive\1. University of Essex\4.0 Secure Software Development\Unit 3>bufoverflow
Enter name: karl
karl
```

What does the output message mean?

When the programme is run for the second time, it is still working with the longer string



```
D:\OneDrive\1. University of Essex\4.0 Secure Software Development\Unit 3>bufoverflow
Enter name: karljackson
karljackson
```

I was expecting to see a buffer overflow error message as the sting is longer than the 8 characters specified. I was also expecting to see the string joined to the previous input as the buffer had not been flushed. However, it was not possible to run the code in the Codio environment as this was not included in the module resources.