Final Exams: Extra Credit Code Snippet
Sabbir Ahmed
Date: December 11, 2017

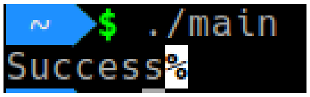
1 More Usefulness of 'printf()'

printf() is one of the most useful functions for beginners in the C programming language. It is a very helpful tool when it comes to debugging, checking statuses of dynamic variables, or simply printing to console. What many users are unaware of is that the function may be used for other very useful tricks. Like the last code snippet, here are some additional tricks with printf().

1.1 %m Specifier

The following snippet will print out "Success".

```
#include <stdio.h>
int main() {
    printf("%m");
    return 0;
}
```



```
~ $ ./main
Success%
```

Figure 1: Running Snippet 1

%m only prints "Success" when "errno == 0". It is short for a string representation of the last observed error state. For instance, if a function fails before the printf(), it will print something rather different. Therefore, the snippet is equivalent to the following line:

```
#include <stdio.h>
int main() {
    printf("%s\n", strerror (errno));
    return 0;
}
```

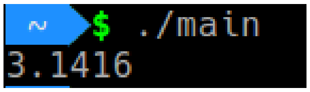
Note that the %m specifier is a GNU C extension of printf().

1.2 '*' Specifier

Another feature with the format specifiers is that the asterisk character (*) may be used to format the specifiers with variables. For example, the following snippet:

```c
#include <stdio.h>
int main() {
    int sig_fig = 4;
    double pi = 3.14159265358979323846;
    printf("%.*f\n", sig_fig, pi);
    return 0;
}
```

will generate the following output:



```
~ $ ./main
3.1416
```

Figure 2: Running Snippet 2

Since `sig_fig` was 4, the `%.*f` specifier printed the value of pi to 4 significant figures.