Lab 1

Robert Krency

```
[kre1188@DRACO1 lab1]$ nano summation.c
[kre1188@DRACO1 lab1]$ gcc summation.c -o summation
[kre1188@DRACO1 lab1]$ ./summation
Summation Calculator
Please enter a nonnegative integer (0 - 2147483647): 261984
Calculating summation for 261984.
The summation of 261984 is 34317939120.
[kre1188@DRACO1 lab1]$ ./summation
Summation Calculator
Please enter a nonnegative integer (0 - 2147483647): 261985
Invalid entry.
Please enter a nonnegative integer (0 - 261984): asdf
Invalid entry.
Please enter a nonnegative integer (0 - 261984): 5
Calculating summation for 5.
The summation of 5 is 15.
[kre1188@DRACO1 lab1]$
```

The apparent limit of recursion from my testing on DRACO1 revealed an upper limit at $n = 261,984$. This is due to the stack limit, as higher values create a stack overflow resulting in a segmentation fault as the recursion depth gets higher. An interative approach would not be limited by this, and would likely be able to calculate any sums up to limits of integer implementations.