Assignment 1
Question 2 Documentation

How Program Works:

1. In the program, we have three C files. One is the **main.c** file, here I have written the **A()** function and made extern void reference to **B.asm** file's **B** function.
2. The A() function takes input for a 64-bit integer value and passes it as an argument to the **B()** function from the **B.asm** file, the B function is called explicitly.
3. The B function first converts the 64-bit integer into an 8byte ASCII string which is then written to the stdout using **write** system calls from x86_64 assembly where the syscall id for write is 1.
4. Then I **mov** the string into the register **rsi** and length in **rdx** and call the syscall to print the ASCII character corresponding to the input integer **(65 -> A)**.
5. Now I have used assembly code to ret from B() to C(), the C() function is in the **C.c** file.
6. The assembly code simply loads the effective memory address using (**lea**) and then moves the address into the **rax** register using the **mov** command.
7. The **rax** is then pushed into the stack using **push**.
8. Then the **ret** command pops the newly added address which is the C() function address and it gets executed and the program **exits**.

Compilation Steps:

1. Firstly open the terminal and cd your way into my submission directory for **Q2**.
2. Now on the terminal type **make**.
3. This file first makes an object file for **B.asm.**
4. Then compiles the program and links **main.c, B.o** and **C.c** into one executable binary by the name **main**.
5. Now just type **./main** onto the terminal to run the program.
6. And finally, input an integer to get its corresponding ASCII code.