1_2

Command to Run

```
@make
```

Make File Description

This line creates the object file using nasm. It takes the B.asm assembly code and converts it to B.o object file

```
@nasm -f elf64 B.asm
```

This line generates the executable named out

```
@gcc A.c B.o C.c -o out -no-pie
```

This line runs the executable

```
@./out
```

Give the input and view the output

Code-

- `A.c`
    - Main Function Calls `A()`
    - `A()` takes Long Input and Calls B. B is added as an extern as it is in an asm file
- `B.asm`
    - rdi is pushed into stack
    - lea just calculates the address, it doesn't actually access memory and stores it into rcx for the value in stack
    - Moving 1 into rax and rdx to denote write syscall and the first parameter
    - Moving rcx into rsi to give the long value
    - syscall is made which writes on stdout the long value as a string
    - mov instruction stores value of C in the rax register which is then pushed
    - Return hence takes to C as rax was pushed on the stack which has C moved in it

- `C.c`
    - `exit(0)` terminates the program