Gcc command-line options that pauses the compilation of the program at each step are:

- **-E** :
    - Stop after the preprocessing stage; do not run the compiler proper.
    - The output is in the form of preprocessed source code, which is sent to the standard output.
    - By default, the pre-processed file name for a source file is made by replacing the suffix .c with .i.
    - Input files that don't require preprocessing are ignored.
    - It also removes comments.

- **-S**
    - Stop after the stage of compilation proper; do not assemble.
    - The output is in the form of an assembler code file for each non-assembler input file specified.
    - By default, the assembler file name for a source file is made by replacing the suffix .c, .i, etc., with .s.
    - Input files that don't require compilation are ignored.

- **-c**
    - Compile or assemble the source files, but do not link.
    - The linking stage simply is not done.
    - The ultimate output is in the form of an object file for each source file.
    - By default, the object file name for a source file is made by replacing the suffix .c, .i, .s, etc., with .o.
    - Unrecognized input files, not requiring compilation or assembly, are ignored.

NASM options used:

- **-felf64** :
    - Specifies the debug information format.
    - Specifies the size of the integer input i.e. 64 bit

Preprocessor file

A preprocessed <filename>.i file is an output of the C or C++ preprocessor.
The preprocessor carries out the following operations:

- Inclusion of header files (the '#include' instruction);
- Macros substitution (the '#define' instruction);
- Conditional compilation (instructions '#if', '#ifdef', '#else', '#elif', '#endif').

I.e. it processes all of the code that is written before the execution of the c program.

Assembly file

These types of files are source code files written in assembly. Assembly is an extremely low-level form of programming. The files contain assembly instructions to the processor in sequential order and are typically compiled based on a selected architecture.
They contain pure assembly code that can be compiled into an object, which the machine can understand.

Object file

A file ending in .o is an object file. It contains compiled object code (that is, machine code produced by the C or C++ compiler), together with the names of the functions and other objects the file contains. Object files are processed by the linker to produce the final executable.

Executable file

An executable file is a type of computer file that runs a program when it is opened. This means it executes code or a series of instructions contained in the file. Executable files have been compiled from source code into binary machine code that is directly executable by the CPU.
An executable file from one operating system will not run on another operating system. This is because the code is executed by the operating system and therefore must be compiled in a format that the operating system can understand.

Working of the Routine:

Header file used : <inttypes.h>

It includes the <stdint.h> header file as well. It is used with printf and scanf functions and also helps us to work with intmax_t type.

ASM Routine:

- It takes the input in 2 registers i.e. rdi & rsi.
- It adds the value of the 2 registers.
- It stores the added value in rax.
- It returns the value in rax

C Routine:

- Add function is declared with int64_t type to tell the compiler that return type is of 64 bit integer.
- 2 integer inputs are taken with the scanf function.
- Add function is called with inputted values
- Returned value is outputted on console using printf function