

# x86 Disassembly

Purpose of this assignment is to reverse engineer x86 assembly language of C code.

```
#include <stdio.h>
#include <stdlib.h>
__attribute__((noinline))
int sum(int a, int b) {
    return a + b;
}
__attribute__((noinline))
void print_the_value(int value) {
    printf("%d", value);
}
int entry_point() {
    int a = rand();
    int b = rand();
    int result = sum(b, a);
    print_the_value(result);}

```

## Problem:

1. Copy and paste the code below to <https://godbolt.org>
2. Use the following compiler for the ASM code generation: `x86-64 gcc 12.2`
3. Under the compiler options, use `-Os`
4. Explain each and every single line of the ASM program
  - You do not have to explain the C code
  - Attempt to provide the WHY rather than the WHAT. For example, do not just say "Moving R1 to R0" but state that "Moving R1 to R0 such that we can pass that as the first parameter to printf"

---

Revision #2

Created 3 years ago by [Preet Kang](#)

Updated 3 years ago by [Preet Kang](#)