# **CS 410 Binary to Assembly Activity Template**

**Step 1:** List the binary file name.

**Step 2:** Identify the functions in the binary file.

**Step 3**: Convert the binary file to assembly code.

**Step 4:** Align the blocks of assembly code with their corresponding function in the binary file.

**Step 5:** Explain the functionality of the blocks of assembly code.

## File One: *assignment3\_1.o*

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| 000000000040057d <main>: | push %rbp | Push value of %rbp to stack. |
| cout | mov $0x400634,%edi callq 0x400450 | Move value at 0x400634 into edi register and printing it |
| cout | mov $0x400648,%edi callq 0x400450 | Move value at 0x400648 into edi register and printing it |
| cout | mov $0x40065c,%edi callq 0x400450 | Moving value at 0x40065c into edi register and printing it |

## File Two: *assignment3\_2.o*

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
|  | push %rbp | Push rdp value to the stack |
|  | mov %rsp,%rbp | Move rsp into rdp |
| Subtract | sub $0x20,%rsp | Subtract 0x20 from rsp |
|  | mov %fs:0x28,%rax | Move fs:0x28 into rax |
|  | mov %rax,-0x8(%rbp) | Move rax into rdp |
| cin | xor %eax,%eax mov $0x400714,%edi callq 4004e0 <puts@plt> | user input |
| Scan for values | lea -0x20(%rbp),%rax mov %rax,%rsi mov $0x40072b,%edi mov $0x0,%eax callq 400520 <\_\_isoc99\_scanf@plt> | Checks for values |
| cout | lea -0x20(%rbp),%rax mov %rax,%rsi mov $0x40072e,%edi mov $0x0,%eax callq 4004fo <printf@plt> | Prints |

## File Three: *assignment3\_3.o*

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
|  | push %rbp | Push rdp value to the stack |
|  | mov %rsp,%rbp | Move rsp into rdp |
| Subtract | sub $0x10,%rsp | Subtract 0x10 from rsp |
| Cin | mov $0x400734,%edi callq 4004e0 <puts@plt> | User Input |
| a,b variables | lea -0x8(%rbp),%rdx lea -0xc(%rbp),%rax | Create the two variables |
| Scan | mov %rax,%rsi mov $0x400747,%edi mov $0x0,%eax callq 400520 <\_\_isoc99\_scanf@plt> | Move rax into rdp, Checks for values |
| add | mov -0x8(%rbp),%edx mov -0xc(%rbp),%eax mov %edx,%esi mov %eax,%edi callq 0x40062d | Add the variables |
| Cout | callq 4004f0 <printf@plt> | Print result |

## File Four: *assignment3\_4.o*

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
|  | push %rbp | Push rdp value to the stack |
|  | mov %rsp,%rbp | Move rsp into rdp |
| Subtract | sub $0x10,%rsp | Subtract 0x10 from rsp |
| Initaiate variable | mov $0x0,-0x8(%rbp) mov -0x8(%rbp),%eax | Creates variable |
| Cin, subtract, compare | mov %eax,%esi mov $0x400844,%edi mov $0x0,%eax callq 4004f0 <printf@plt> subl $0x1,-0x8(%rbp) cmpl $0x0,-0x8(%rbp) | Move eax into esi then prints the variable and then subtract one and then check if the number is 1 |
| For loop | jg 0x400647 mov -0x4(%rbp),%eax mov %eax,%esi mov $0x400848,%edi mov $0x0,%eax callq 0x4004f0 mov -0x4(%rbp),%eax leaveq retq | Prints, then loop until value is 1 |