# **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** |
| --- | --- | --- |
| Main | Push %rbp | Push stack to rbp |
|  | Mov %rsp, %rbp | Move contents of rbp to rsp |
|  | Mov $0x400634,%edi | Move contents to edi |
|  | callq 0x400450 <puts@plt> | Function call at address |
|  | mov $0x400648,%edi | Move contents to edi |
|  | callq 0x400450 <puts@plt> | Function call |
|  | mov $0x40065c,%edi | Move contents to edi |
|  | callq 0x400450 <puts@plt> | Function call |
|  | mov $0x0,%edi | Move contents to edi |
|  | callq 0x400480 <exit@plt> | Exit function |

## File Two: assignment3\_2.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Push stack to rbp |
|  | mov %rsp,%rbp | Mov rbp to rsp |
|  | sub $0x20,%rsp | Subtract value from rsp |
|  | mov %fs:0x28,%rax | Move contents to rax |
|  | mov %rax,-0x8(%rbp) | Move rax to 8above rbp |
|  | xor %eax,%eax | Or operation on eax |
|  | mov $0x400714,%edi | Move contents to edi |
|  | callq 0x4004e0 <puts@plt> | Fucntion call |
|  | lea -0x20(%rbp),%rax | Load address 20 rbp to rax |
|  | mov %rax,%rsi | Move rsi to rax |
|  | mov $0x40072b,%edi | Mov contents to edi |
|  | mov $0x0,%eax | Move 0 to eax |
|  | callq 0x400520 <\_\_isoc99\_scanf@plt> | Scans for matching value |
|  | lea -0x20(%rbp),%rax | Load address rax to 20 rbp |
|  | mov %rax,%rsi | Move rsi to rax |
|  | mov $0x40072e,%edi | Move value to edi |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x4004f0 <printf@plt> | Call print function |
|  | mov $0x0,%edi | Move zero to edi |
|  | callq 0x400530 <exit@plt> | Exit function |

## File Three: assignment3\_3.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Push stack to rbp |
|  | mov %rsp,%rbp | Move rbp to rsp |
|  | sub $0x10,%rsp | Subtract 10 from rsp |
|  | mov $0x400734,%edi | Move content to edi |
|  | callq 0x4004e0 <puts@plt> | Call function |
|  | lea -0x8(%rbp),%rdx | Load address rdx to 8 rbp |
|  | lea -0xc(%rbp),%rax | Load address rax to C rbp |
|  | mov %rax,%rsi | Move rsi to rax |
|  | mov $0x400747,%edi | Move value to edi |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x400520 <\_\_isoc99\_scanf@plt> | Call scan to check match |
|  | mov -0x8(%rbp),%edx | Move ex to 8 rbp |
|  | mov -0xc(%rbp),%eax | Move eax to c rbp |
|  | mov %edx,%esi | Move esi to edx |
|  | mov %eax,%edi | Move edi to eax |
|  | callq 0x40062d <AddNumbers> | Call addnumbers function |
|  | mov %eax,-0x4(%rbp) | Move 4rbp to eax |
|  | mov -0x8(%rbp),%edx | Move edx to 8 rbp |
|  | mov -0xc(%rbp),%eax | Move eax to c rbp |
|  | mov -0x4(%rbp),%ecx | Move ecx to 4 rbp |
|  | mov %eax,%esi | Move esi to eax |
|  | mov $0x40074d,%edi | Move value to edi |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x4004f0 <printf@plt> | Call print function |
|  | mov $0x0,%edi | Move zero to edi |
|  | callq 0x400530 <exit@plt> | Exit function |

## File Four: assignment3\_4.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Push stack to rbp |
|  | mov %rsp,%rbp | Move rbp to rsp |
|  | sub $0x10,%rsp | Subtract 10 from rsp |
|  | movl $0x0,-0x8(%rbp) | Move zero to 8 rbp |
|  | jmp 0x4007a0 <main+137> | Jump to 137 of main |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x4006df <DisplayMenu> | Call function display menu |
|  | mov $0x400886,%edi | Move value to edi |
|  | callq 0x4004e0 <puts@plt> | Call function |
|  | lea -0x8(%rbp),%rax | Load address rax to 8 rbp |
|  | mov %rax,%rsi | Move rsi to rax |
|  | mov $0x400899,%edi | Move value to edi |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x400520 <\_\_isoc99\_scanf@plt> | Call scan function to check for match |
|  | mov -0x8(%rbp),%eax | Move eax to 8 rbp |
|  | cmp $0x3,%eax | Compare value against eax |
|  | je 0x40077a <main+99> | Jump to 99 of main if compare is equal |
|  | mov $0x40089c,%edi | Move value to edi |
|  | callq 0x4004e0 <puts@plt> | Call the function |
|  | lea -0x4(%rbp),%rax | Load rax to 4 rbp |
|  | mov %rax,%rsi | Move rsi to rax |
|  | mov $0x400899,%edi | Move the value to edi |
|  | mov $0x0,%eax | Move zero to eax |
|  | callq 0x400520 <\_\_isoc99\_scanf@plt> | Call scan to check for match |
|  | mov -0x8(%rbp),%eax | Move eax to 8 rbp |
|  | cmp $0x1,%eax | Comp eax against value |
|  | jne 0x40078e <main+119> | Jump to main 119 if above not equal |
|  | mov -0x4(%rbp),%eax | Move eax tp 4 rbp |
|  | mov %eax,%edi | Move edi to eax |
|  | callq 0x40062d <PrintFact> | Call PrintFact Function |
|  | jmp 0x4007a0 <main+137> | Jump to main 137 |
|  | mov -0x8(%rbp),%eax | Move eax to 8 rbp |
|  | cmp $0x2,%eax | Compare eax to value |
|  | jne 0x4007a0 <main+137> | Jump to 137 main if not equal |
|  | mov -0x4(%rbp),%eax | Move eax to 4rbp |
|  | mov %eax,%edi | Move edi to eax |
|  | callq 0x400688 <PrintSum> | Call PrintSum fucntion |
|  | mov -0x8(%rbp),%eax | Move eax to 8 rbp |
|  | cmp $0x3,%eax | Compare eax to value |
|  | jne 0x400728 <main+17> | Jump to main 17 if not equal |
|  | mov $0x0,%edi | Move zero to edi |
|  | callq 0x400530 <exit@plt> | Exit function |