# **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 | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| mov $0x400634,%edi | Moves and Assigns $0x400634 into %edi |
| callq 400450 <puts@plt> | Calls and outputs 400450 |
| mov $0x400648,%edi | Moves and Assigns $0x400648 into %edi |
| callq 400450 <puts@plt> | Calls and outputs 400450 |
| mov $0x40065c,%edi | Moves and Assigns $0x40065c into %edi |
| callq 400450 <puts@plt> | Calls and outputs 400450 |
| mov $0x0,%edi | Moves and Assigns $0x40065c into %edi |
| callq 400480 <exit@plt> | Calls exit function |
| nopl 0x0(%rax) | Exit main function |

## File Two: assignment3\_2.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Registers %rbp as base |
|  | mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
|  | sub $0x20,%rs | Subtracts $0x20 from %rs |
|  | mov %fs:0x28,%rax | Moves and Assigns %fs:0x28 into %rax |
|  | mov %rax,-0x8(%rbp) | Moves and Assigns %rax into -0x8(%rbp) |
|  | xor %eax,%eax | Compares %eax and generates first output |
|  | mov $0x400714,%edi | Moves and Assigns %rax into -0x8(%rbp) |
|  | callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
|  | lea -0x20(%rbp),%rax | Reads -0x20(%rbp) and loads into %rax |
|  | mov %rax,%rsi | Moves and Assigns %rax into %rsi |
|  | mov $0x40072b,%edi | Moves and Assigns $0x40072b into %edi |
|  | mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
|  | callq 400520 <\_\_isoc99\_scanf@plt> | Calls 400520 and assign into -0x20(%rbp)s |
|  | lea -0x20(%rbp),%rax | Reads -0x20(%rbp) and loads into %rax |
|  | mov %rax,%rsi | Moves and Assigns %rax into % rsi |
|  | mov $0x40072e,%edi | Moves and Assigns $0x40072e into %edi |
|  | mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
|  | callq 4004f0 <printf@plt> | Calls printf function |
|  | mov $0x0,%edi | Moves and Assigns $0x0 into %edi |
|  | callq 400530 <exit@plt> | Calls exit function |

## File Three: assignment3\_3.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into $rbp |
| sub $0x10,%rsp | Subtracts $0x10 from %rsp |
| mov $0x400734,%edi | Moves and Assigns $0x400734 into %di |
| callq 4004e0 <puts@plt> | Calls and output 4004e0 |
| lea -0x8(%rbp),%rdx | Reads -0x8(%rbp) and loads into %rax |
| lea -0xc(%rbp),%rax | Reads -0xc(%rbp ) and loads into %rax |
| mov %rax,%rsi | Moves and Assigns %rax into %rsi |
| mov $0x400747,%edi | Moves and Assigns $0x400747 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 400520 <\_\_isoc99\_scanf@plt> | Calls 400520 and output -0xc(%rbp ) |
| mov -0x8(%rbp),%edx | Moves and Assigns -0x8(%rbp) into %edx |
| mov -0xc(%rbp),%eax | Moves and Assigns -0xc(%rbp) into %eax |
| mov %edx,%esi | Moves and Assigns %edx into %esi |
| mov %eax,%edi | Moves and Assigns %eax into %edi |
| callq 40062d <AddNumbers> | Calls the AddNumber function |
| mov %eax,-0x4(%rbp) | Moves and Assigns %eax into -0x4(%rbp) |
| mov -0x8(%rbp),%edx | Moves and Assigns -0x8(%rbp) into %edx |
| mov -0xc(%rbp),%eax | Moves and Assigns -0xc(%rbp) into %eax |
| mov -0x4(%rbp),%ecx | Moves and Assigns -0x4(%rbp) into %ecx |
| mov %eax,%esi | Moves and Assigns %eax into %esi |
| mov $0x40074d,%edi | Moves and Assigns $0x40074d into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4004f0 <printf@plt> | Calls printf funtion |
| mov $0x0,%edi | Moves and Assigns $0x0 into %edi |
| callq 400530 <exit@plt> | Calls exit function |
| AddNumbers | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| mov %edi,-0x4(%rbp) | Moves and Assigns %edi into -0x4(%rbp) |
| mov %esi,-0x8(%rbp) | Moves and Assigns %esi into -0x8(%rbp) |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| mov -0x4(%rbp),%edx | Moves and Assigns -0x4(%rbp) into %edx |
| add %edx,%eax | Adds %edx and %eax |
| pop %rbp | Retrieves and stores into %rbp |
| retq | Return and continue execution |

## File Four: assignment3\_4.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| Main | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| sub $0x10,%rsp | Subtract $0x10 from %rsp |
| movl $0x0,-0x8(%rbp) | Moves 4 bytes of $0x0 and assigns into -0x8(%rbp) |
| jmp 4007a0 <main+0x89> | Jump to main function |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4006df <DisplayMenu> | Calls DisplayMenu Function |
| mov $0x400886,%edi | Moves and Assigns $0x400886 into %edi |
| callq 4004e0 <puts@plt> | Calls and outpus 4004e0 |
| lea -0x8(%rbp),%rax | Reads -0x8(%rbp ) and loads into %rax |
| mov %rax,%rsi | Moves and Assigns %rax into %rsi |
| mov $0x400899,%edi | Moves and Assigns $0x400899 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 400520 <\_\_isoc99\_scanf@plt> | Calls 400520 and output -0x8(%rbp ) |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| cmp $0x3,%eax | Compares the values of $0x3 with %eax |
| je 40077a <main+0x63> | IF value is equal, jump to Main |
| mov $0x40089c,%edi | Moves $0x40089c into %edi |
| callq 4004e0 <puts@plt> | Calls and outpus 4004e0 |
| lea -0x4(%rbp),%rax | Reads -0x4(%rbp ) and loads into %rax |
| mov %rax,%rsi | Moves and Assigns %rax into %rsi |
| mov $0x400899,%edi | Moves and Assigns $0x400899 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 400520 <\_\_isoc99\_scanf@plt> | Calls 400520 and output -0x8(%rbp ) |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| cmp $0x1,%eax | Compares the values of $0x1 with %eax |
| jne 40078e <main+0x77> | IF value is not equal, jump to Main |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| mov %eax,%edi | Moves and Assigns %eax into %edi |
| callq 40062d <PrintFact> | Calls the PrintFact function |
| jmp 4007a0 <main+0x89> | Jump to main function |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| cmp $0x2,%eax | Compares the values of $0x2 with %eax |
| jne 4007a0 <main+0x89> | IF value is not equal, jump to Main |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| mov %eax,%edi | Moves and Assigns %eax into %edi |
| callq 400688 <PrintSum> | Calls the PrintSum function |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| cmp $0x3,%eax | Compares the values of $0x3 with %eax |
| jne 400728 <main+0x11> | IF value is not equal, jump to Main |
| mov $0x0,%edi | Moves and Assigns $0x0 into %edi |
| callq 400530 <exit@plt> | Calls exit function |
| DisplayMenu | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| mov $0x400851,%edi | Moves and Assigns $0x400851 into %edi |
| callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
| mov $0x400864,%edi | Moves and Assigns $0x400864 into %edi |
| callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
| mov $0x400871,%edi | Moves and Assigns $0x400871 into %edi |
| callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
| mov $0x40087e,%edi | Moves and Assigns $0x40087e into %edi |
| callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
| mov $0x400851,%edi | Moves and Assigns $0x400851 into %edi |
| callq 4004e0 <puts@plt> | Calls and outputs 4004e0 |
| pop %rbp | Retrieves and stores into %rbp |
| retq | Return and continue execution |
| PrintSum | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| sub $0x20,%rsp | Subtracts $0x20 from %rsp |
| mov %edi,-0x14(%rbp) | Moves and Assigns %edi into -0x14(%rbp) |
| movl $0x0,-0x4(%rbp) | Moves 4 bytes of $0x0 into -0x4(%rbp) |
| mov -0x14(%rbp),%eax | Moves and Assigns -0x14(%rbp) into %eax |
| mov %eax,-0x8(%rbp) | Moves and Assigns %eax into -0x8(%rbp) |
| jmp 4006c0 <PrintSum+0x38> | Jump to PrintsSum function |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| add %eax,-0x4(%rbp) | Adds %eax and -0x4(%rbp) |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| mov %eax,%esi | Moves and Assigns %eax into %esi |
| mov $0x400844,%edi | Moves and Assigns $0x400844 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4004f0 <printf@plt> | Calls the printf function |
| subl $0x1,-0x8(%rbp) | Subtract $0x1 from -0x8(%rbp) |
| cmpl $0x0,-0x8(%rbp) | Compares $0x0 and -0x8(%rbp) |
| jg 4006a2 <PrintSum+0x1a> | Jumps to PrintSum IF greater |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| mov %eax,%esi | Moves and Assigns %eax into %esi |
| mov $0x400848,%edi | Moves and Assigns $0x400848 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4004f0 <printf@plt> | Calls the printf function |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| leaveq | Cleans stack and prepares exit |
| retq | Return and continue execution |
| PrintFact | push %rbp | Registers %rbp as base |
| mov %rsp,%rbp | Moves and Assigns %rsp into %rbp |
| sub $0x20,%rsp | Subtracts $0x20 from %rsp |
| mov %edi,-0x14(%rbp) | Moves and Assigns %edi into -0x14(%rbp) |
| movl $0x1,-0x4(%rbp) | Moves and Assigns 4 bytes of $0x1 into -0x4(%rbp) |
| mov -0x14(%rbp),%eax | Moves and Assigns -0x14(%rbp) into %eax |
| mov %eax,-0x8(%rbp) | Moves %eax into -0x8(%rbp) |
| jmp 400669 <PrintFact+0x3c> | Jump to PrintFact function |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| imul -0x8(%rbp),%eax | Multilply -0x8(%rbp) by %eax |
| mov %eax,-0x4(%rbp) | Moves and Assigns %eax into -0x4(%rbp) |
| mov -0x8(%rbp),%eax | Moves and Assigns -0x8(%rbp) into %eax |
| mov %eax,%esi | Moves and Assigns %eax into %esi |
| mov $0x400844,%edi | Moves and Assigns $0x400844 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4004f0 <printf@plt> | Calls printf function |
| subl $0x1,-0x8(%rbp) | Subtracts $0x1 from -0x8(%rbp) |
| cmpl $0x0,-0x8(%rbp) | Compares $0x0 and -0x8(%rbp) |
| jg 400647 <PrintFact+0x1a> | Jumps to PrintFact IF greater |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| mov %eax,%esi | Moves and Assigns %eax into %esi |
| mov $0x400848,%edi | Moves and Assigns $0x400848 into %edi |
| mov $0x0,%eax | Moves and Assigns $0x0 into %eax |
| callq 4004f0 <printf@plt> | Calls the printf function |
| mov -0x4(%rbp),%eax | Moves and Assigns -0x4(%rbp) into %eax |
| leaveq | Cleans stack and prepares exit |
| retq | Return and continue execution |