# **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** |
| --- | --- | --- |
| 0x000000000040057d main |  | Main of the program |
| 0x000000000040057d <+0>: | push %rbp | Stores a value |
| 0x000000000040057e <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x0000000000400581 <+4>: | mov $0x400634,%edi | Moves a unique string to %edi |
| 0x0000000000400586 <+9>: | callq 0x400450 <puts@plt> | Calls print string |
| 0x000000000040058b <+14>: | mov $0x400648,%edi | Moves a second unique string into %edi |
| 0x0000000000400590 <+19>: | callq 0x400450 <puts@plt> | Calls print string |
| 0x0000000000400595 <+24>: | mov $0x40065c,%edi | Moves a third unique string into %edi |
| 0x000000000040059a <+29>: | callq 0x400450 <puts@plt> | Calls print string |
| 0x000000000040059f <+34>: | mov $0x0,%edi | Moves 0 into %edi |
| 0x00000000004005a4 <+39>: | callq 0x400480 <exit@plt> | Exits the program |
|  |  | This program calls three unique strings and prints them out into the terminal |

## File Two: Assignment3\_2.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| 0x000000000040062d main |  | Main of the program |
| 0x000000000040062d <+0>: | push %rbp | Stores a value |
| 0x000000000040062e <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x0000000000400631 <+4>: | sub $0x20,%rsp | Subtracts 20 from %rsp |
| 0x0000000000400635 <+8>: | mov %fs:0x28,%rax | Move 28 to %rax |
| 0x000000000040063e <+17>: | mov %rax,-0x8(%rbp) | Move %rax to -0x8(%rbp) |
| 0x0000000000400642 <+21>: | xor %eax,%eax | Assign 0 value of %eax to %eax |
| 0x0000000000400644 <+23>: | mov $0x400714,%edi | Moves $0x400714 into %eax |
| 0x0000000000400649 <+28>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x000000000040064e <+33>: | lea -0x20(%rbp),%rax | Places -0x20(%rbp) into the register %rax |
| 0x0000000000400652 <+37>: | mov %rax,%rsi | Moves the value in %rax into %rsi |
| 0x0000000000400655 <+40>: | mov $0x40072b,%edi | Moves $0x40072b into %edi |
| 0x000000000040065a <+45>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x000000000040065f <+50>: | callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf |
| 0x0000000000400664 <+55>: | lea -0x20(%rbp),%rax | Places -0x20(%rbp) into the register %rax |
| 0x0000000000400668 <+59>: | mov %rax,%rsi | Moves the value in %rax into %rsi |
| 0x000000000040066b <+62>: | mov $0x40072e,%edi | Moves $0x40072e into %edi |
| 0x0000000000400670 <+67>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x0000000000400675 <+72>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x000000000040067a <+77>: | mov $0x0,%edi | Moves 0 into %edi |
| 0x000000000040067f <+82>: | callq 0x400530 <exit@plt> | Exits the program |
|  |  | This program prompts the user to input a value for their name, after doing this it stores it then prints it back out in the terminal |

## File Three: Assignment3\_3.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| 0x000000000040062d AddNumbers |  | AddNumbers portion of the program |
| 0x000000000040062d <+0>: | push %rbp | Stores a value |
| 0x000000000040062e <+1>: | mov %rsp,%rbp | Moves the value of %rsp into %rbp |
| 0x0000000000400631 <+4>: | mov %edi,-0x4(%rbp) | Moves the value of %edi into -0x4(%rbp) |
| 0x0000000000400634 <+7>: | mov %esi,-0x8(%rbp) | Moves the value of %esi into -0x8(%rbp) |
| 0x0000000000400637 <+10>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x000000000040063a <+13>: | mov -0x4(%rbp),%edx | Moves the value of -0x4(%rbp) into %edx |
| 0x000000000040063d <+16>: | add %edx,%eax | Adds the values stored in %edx and %eax |
| 0x000000000040063f <+18>: | pop %rbp | Restores the value of %rbp |
| 0x0000000000400640 <+19>: | retq | Returns to main |
|  |  |  |
| 0x0000000000400641 main |  | Main portion of the program |
| 0x0000000000400641 <+0>: | push %rbp | Stores a value |
| 0x0000000000400642 <+1>: | mov %rsp,%rbp | Moves the values of %rsp into %rbp |
| 0x0000000000400645 <+4>: | sub $0x10,%rsp | Subtracts 10 from %rsp |
| 0x0000000000400649 <+8>: | mov $0x400734,%edi | Moves $0x400734 into %edi |
| 0x000000000040064e <+13>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x0000000000400653 <+18>: | lea -0x8(%rbp),%rdx | Places -0x8(%rbp) into the register %rdx |
| 0x0000000000400657 <+22>: | lea -0xc(%rbp),%rax | Places -0xc(%rbp) into the register %rax |
| 0x000000000040065b <+26>: | mov %rax,%rsi | Moves the value held by %rax into %rsi |
| 0x000000000040065e <+29>: | mov $0x400747,%edi | Moves $0x400747 into %edi |
| 0x0000000000400663 <+34>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x0000000000400668 <+39>: | callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf |
| 0x000000000040066d <+44>: | mov -0x8(%rbp),%edx | Moves -0x8(%rbp) into %edx |
| 0x0000000000400670 <+47>: | mov -0xc(%rbp),%eax | Moves -0xc(%rbp) into %eax |
| 0x0000000000400673 <+50>: | mov %edx,%esi | Moves the value held by %edx into %esi |
| 0x0000000000400675 <+52>: | mov %eax,%edi | Moves the value held by %eax into %edi |
| 0x0000000000400677 <+54>: | callq 0x40062d <AddNumbers> | Calls the function AddNumbers |
| 0x000000000040067c <+59>: | mov %eax,-0x4(%rbp) | Moves the value held by %eax into ,-0x4(%rbp) |
| 0x000000000040067f <+62>: | mov -0x8(%rbp),%edx | Moves -0x8(%rbp) into %edx |
| 0x0000000000400682 <+65>: | mov -0xc(%rbp),%eax | Moves -0xc(%rbp) into %eax |
| 0x0000000000400685 <+68>: | mov -0x4(%rbp),%ecx | Moves -0x4(%rbp) into %ecx |
| 0x0000000000400688 <+71>: | mov %eax,%esi | Moves the value held by %eax into %esi |
| 0x000000000040068a <+73>: | mov $0x40074d,%edi | Moves $0x40074d into %edi |
| 0x000000000040068f <+78>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x0000000000400694 <+83>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x0000000000400699 <+88>: | mov $0x0,%edi | Moves 0 into %edi |
| 0x000000000040069e <+93>: | callq 0x400530 <exit@plt> | Exits the Program |
|  |  | This program takes in two inputs from the user and stores them, they are later called into the function AddNumber in order to add them and then returns the value back to our main and prints it out. |

## File Four: Assignment3\_4.o

| **Functions** | **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- | --- |
| 0x000000000040062d PrintFact |  | PrintFact section of our program |
| 0x000000000040062d <+0>: | push %rbp | Stores a value |
| 0x000000000040062e <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x0000000000400631 <+4>: | sub $0x20,%rsp | Subtracts 20 from %rsp |
| 0x0000000000400635 <+8>: | mov %edi,-0x14(%rbp) | Moves the value held by %edi into -0x14(%rbp) |
| 0x0000000000400638 <+11>: | movl $0x1,-0x4(%rbp) | Moves $0x1 into -0x4(%rbp) |
| 0x000000000040063f <+18>: | mov -0x14(%rbp),%eax | Moves -0x14(%rbp) into %eax |
| 0x0000000000400642 <+21>: | mov %eax,-0x8(%rbp) | Moves %eax into -0x8(%rbp) |
| 0x0000000000400645 <+24>: | jmp 0x400669 <PrintFact+60> | Jumps to PrintFact+60 |
| 0x0000000000400647 <+26>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x000000000040064a <+29>: | imul -0x8(%rbp),%eax | Multiplies -0x8(%rbp) and %eax |
| 0x000000000040064e <+33>: | mov %eax,-0x4(%rbp) | Moves the value held by %eax into ,-0x4(%rbp) |
| 0x0000000000400651 <+36>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x0000000000400654 <+39>: | mov %eax,%esi | Moves the value held by %eax into %esi |
| 0x0000000000400656 <+41>: | mov $0x400844,%edi | Moves $0x400844 into %edi |
| 0x000000000040065b <+46>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x0000000000400660 <+51>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x0000000000400665 <+56>: | subl $0x1,-0x8(%rbp) | Subtracts $0x1 from -0x8(%rbp) |
| 0x0000000000400669 <+60>: | cmpl $0x0,-0x8(%rbp) | Compares the contents of 0 and -0x8(%rbp) |
| 0x000000000040066d <+64>: | jg 0x400647 <PrintFact+26> | Jumps if the destination operand is greater than the source operand |
| 0x000000000040066f <+66>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x0000000000400672 <+69>: | mov %eax,%esi | Moves the value held by %eax into %esi |
| 0x0000000000400674 <+71>: | mov $0x400848,%edi | Moves 400848 to %edi |
| 0x0000000000400679 <+76>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x000000000040067e <+81>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x0000000000400683 <+86>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x0000000000400686 <+89>: | leaveq | Restores the stack frame |
| 0x0000000000400687 <+90>: | retq | Returns to main |
|  |  |  |
| 0x0000000000400688 PrintSum |  | PrintSum portion of our program |
| 0x0000000000400688 <+0>: | push %rbp |  |
| 0x0000000000400689 <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x000000000040068c <+4>: | sub $0x20,%rsp | Subtracts 20 from %rsp |
| 0x0000000000400690 <+8>: | mov %edi,-0x14(%rbp) | Moves the value held by %edi into -0x14(%rbp) |
| 0x0000000000400693 <+11>: | movl $0x0,-0x4(%rbp) |  |
| 0x000000000040069a <+18>: | mov -0x14(%rbp),%eax | Moves -0x14(%rbp) into %eax |
| 0x000000000040069d <+21>: | mov %eax,-0x8(%rbp) | Moves %eax into -0x8(%rbp) |
| 0x00000000004006a0 <+24>: | jmp 0x4006c0 <PrintSum+56> | Jumps to PrintSum+56 |
| 0x00000000004006a2 <+26>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x00000000004006a5 <+29>: | add %eax,-0x4(%rbp) | Adds %eax and -0x4(%rbp) |
| 0x00000000004006a8 <+32>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x00000000004006ab <+35>: | mov %eax,%esi | Moves the value held by %eax into %esi |
| 0x00000000004006ad <+37>: | mov $0x400844,%edi | Moves $0x400844 into %edi |
| 0x00000000004006b2 <+42>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x00000000004006b7 <+47>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x00000000004006bc <+52>: | subl $0x1,-0x8(%rbp) | Subtracts $0x1 from -0x8(%rbp) |
| 0x00000000004006c0 <+56>: | cmpl $0x0,-0x8(%rbp) | Compares the contents of 0 and -0x8(%rbp) |
| 0x00000000004006c4 <+60>: | jg 0x4006a2 <PrintSum+26> | Jumps if the destination operand is greater than the source operand |
| 0x00000000004006c6 <+62>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x00000000004006c9 <+65>: | mov %eax,%esi | Moves the value held by %eax into %esi |
| 0x00000000004006cb <+67>: | mov $0x400848,%edi | Moves 400848 to %edi |
| 0x00000000004006d0 <+72>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x00000000004006d5 <+77>: | callq 0x4004f0 <printf@plt> | Calls print string |
| 0x00000000004006da <+82>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x00000000004006dd <+85>: | leaveq | Restores the stack frame |
| 0x00000000004006de <+86>: | retq | Returns to main |
|  |  |  |
| 0x00000000004006df DisplayMenu |  | DisplayMenu portion of our Program |
| 0x00000000004006df <+0>: | push %rbp |  |
| 0x00000000004006e0 <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x00000000004006e3 <+4>: | mov $0x400851,%edi | Moves $0x400851into %edi |
| 0x00000000004006e8 <+9>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x00000000004006ed <+14>: | mov $0x400864,%edi | Moves $0x400864 into %edi |
| 0x00000000004006f2 <+19>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x00000000004006f7 <+24>: | mov $0x400871,%edi | Moves $0x400871 into %edi |
| 0x00000000004006fc <+29>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x0000000000400701 <+34>: | mov $0x40087e,%edi | Moves $0x40087e into %edi |
| 0x0000000000400706 <+39>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x000000000040070b <+44>: | mov $0x400851,%edi | Moves $0x400851into %edi |
| 0x0000000000400710 <+49>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x0000000000400715 <+54>: | pop %rbp | Restores the value of %rbp |
| 0x0000000000400716 <+55>: | retq | Returns to main |
|  |  |  |
| 0x0000000000400717 main |  | Main portion of our porgram |
| 0x0000000000400717 <+0>: | push %rbp |  |
| 0x0000000000400718 <+1>: | mov %rsp,%rbp | Moves the value held by %rsp into %rbp |
| 0x000000000040071b <+4>: | sub $0x10,%rsp |  |
| 0x000000000040071f <+8>: | movl $0x0,-0x8(%rbp) |  |
| 0x0000000000400726 <+15>: | jmp 0x4007a0 <main+137> | Jumps to main+137 |
| 0x0000000000400728 <+17>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x000000000040072d <+22>: | callq 0x4006df <DisplayMenu> | Calls the DisplayMenu Function |
| 0x0000000000400732 <+27>: | mov $0x400886,%edi | Moves $0x400886 into %edi |
| 0x0000000000400737 <+32>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x000000000040073c <+37>: | lea -0x8(%rbp),%rax | Places -0x8(%rbp) into the register %rax |
| 0x0000000000400740 <+41>: | mov %rax,%rsi | Moves the value in %rax into %rsi |
| 0x0000000000400743 <+44>: | mov $0x400899,%edi | Moves $0x400899 into %edi |
| 0x0000000000400748 <+49>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x000000000040074d <+54>: | callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf |
| 0x0000000000400752 <+59>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x0000000000400755 <+62>: | cmp $0x3,%eax | Compares the contents of $0x3 and %eax |
| 0x0000000000400758 <+65>: | je 0x40077a <main+99> | Jumps only if 0x40077a is equal |
| 0x000000000040075a <+67>: | mov $0x40089c,%edi | Moves $0x40089c into %edi |
| 0x000000000040075f <+72>: | callq 0x4004e0 <puts@plt> | Calls print string |
| 0x0000000000400764 <+77>: | lea -0x4(%rbp),%rax | Places -0x4(%rbp) into the register %rax |
| 0x0000000000400768 <+81>: | mov %rax,%rsi | Moves the value in %rax into %rsi |
| 0x000000000040076b <+84>: | mov $0x400899,%edi | Moves $0x400899 into %edi |
| 0x0000000000400770 <+89>: | mov $0x0,%eax | Moves 0 into %eax |
| 0x0000000000400775 <+94>: | callq 0x400520 <\_\_isoc99\_scanf@plt> | Calls scanf |
| 0x000000000040077a <+99>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x000000000040077d <+102>: | cmp $0x1,%eax | Compares the contents of $0x1 and %eax |
| 0x0000000000400780 <+105>: | jne 0x40078e <main+119> | Jumps if 0x40078e is not equal |
| 0x0000000000400782 <+107>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x0000000000400785 <+110>: | mov %eax,%edi | Moves the value held by %eax into %edi |
| 0x0000000000400787 <+112>: | callq 0x40062d <PrintFact> | Calls the PrintFact Function |
| 0x000000000040078c <+117>: | jmp 0x4007a0 <main+137> | Jumps to main+137 |
| 0x000000000040078e <+119>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x0000000000400791 <+122>: | cmp $0x2,%eax | Compares the contents of $0x2 and %eax |
| 0x0000000000400794 <+125>: | jne 0x4007a0 <main+137> | Jumps if 0x4007a0 is not equal |
| 0x0000000000400796 <+127>: | mov -0x4(%rbp),%eax | Moves -0x4(%rbp) into %eax |
| 0x0000000000400799 <+130>: | mov %eax,%edi | Moves the value held by %eax into %edi |
| 0x000000000040079b <+132>: | callq 0x400688 <PrintSum> | Calls the PrintSum Function |
| 0x00000000004007a0 <+137>: | mov -0x8(%rbp),%eax | Moves the value of -0x8(%rbp) into %eax |
| 0x00000000004007a3 <+140>: | cmp $0x3,%eax | Compares the contents of $0x3 and %eax |
| 0x00000000004007a6 <+143>: | jne 0x400728 <main+17> | Jumps if 0x400728 is not equal |
| 0x00000000004007a8 <+145>: | mov $0x0,%edi | Moves 0 into %edi |
| 0x00000000004007ad <+150>: | callq 0x400530 <exit@plt> | Exits the Program |
|  |  | This program has 3 functions they are: Print Factorial, Print Sum, and Display the menu based off the input that the user enter it can take you to any of these three option which will then call the necessary function do the needed operation print the results and then return to main. |