CS 410 Binary to Assembly Activity

File One: assignment3_1.o		
Functions	**Blocks of Assembly Code**	**Explanation of Functionality**
main	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	mov $0x400634,%edi	Assigns value of $0x400634 to %edi and outputs the value
	callq 0x400450 <puts@plt>	
	mov $0x400648,%edi	Assigns value of $0x400648 to %edi and outputs the value
	callq 0x400450 <puts@plt>	
	mov $0x40065c,%edi	Assigns value of $0x400634 to %edi and outputs the value
	callq 0x400450 <puts@plt>	
	mov $0x0,%edi	Exits the main function and ends the program
	callq 0x400480 <exit@plt>	

File Two: assignment3_2.o		
Functions	**Blocks of Assembly Code**	**Explanation of Functionality**
main	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	sub $0x20,%rsp	Subtracts the value of $0x20 from %rsp
	mov %fs:0x28,%rax	Assigns value from %rax to -0x8(%rbp)
	mov %rax,-0x8(%rbp)	
	xor %eax,%eax	Conditionally outputs value
	mov $0x400714,%edi	
	callq 0x4004e0 <puts@plt>	
	lea -0x20(%rbp),%rax	Reads in a value and assigns it to the location -0x20(%rbp)
	mov %rax,%rsi	
	mov $0x40072b,%edi	
	mov $0x0,%eax	
	callq 0x400520 <__isoc99_scanf@plt>	
	lea -0x20(%rbp),%rax	Outputs the value from -0x20(%rbp)
	mov %rax,%rsi	
	mov $0x40072e,%edi	
	mov $0x0,%eax	
	callq 0x4004f0 <printf@plt>	
	mov $0x0,%edi	Exits the main function and ends the program
	callq 0x400530 <exit@plt>	

File Three: assignment3_3.o		
Functions	**Blocks of Assembly Code**	**Explanation of Functionality**
main	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	sub $0x10,%rsp	Subtracts the value of $0x10 from %rsp
	mov $0x400734,%edi	Assigns value of $0x400734 to %edi
	callq 0x4004e0 <puts@plt>	Reads in value and prints out statement
	lea -0x8(%rbp),%rdx	
	lea -0xc(%rbp),%rax	
	mov %rax,%rsi	
	mov $0x400747,%edi	
	mov $0x0,%eax	
	callq 0x400520 <__isoc99_scanf@plt>	
	mov -0x8(%rbp),%edx	Assign -0x8(%rbp) to %edx
	mov -0xc(%rbp),%eax	Assign -0xc(%rbp) to %eax
	mov %edx,%esi	Move %edx to %esi
	mov %eax,%edi	Move %eax to %edi
	callq 0x40062d <AddNumbers>	Calls AddNumbers function
	mov %eax,-0x4(%rbp)	Move %eax to -0x4(%rbp)
	mov -0x8(%rbp),%edx	Assign -0x8(%rbp) to %edx
	mov -0xc(%rbp),%eax	Assign -0xc(%rbp) to %eax
	mov -0x4(%rbp),%ecx	Assign -0x4(%rbp) to %ecx
	mov %eax,%esi	Move %eax to %esi
	mov $0x40074d,%edi	Assign $0x40074d to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4004f0 <printf@plt>	Display output
	mov $0x0,%edi	Exits the main function and ends the program
	callq 0x400530 <exit@plt>	
AddNumbers	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	mov %edi,-0x4(%rbp)	Assign value of %edi to -0x4(rbp)
	mov %esi,-0x8(%rbp)	Assign value of %esi to -0x8(rbp)
	mov -0x8(%rbp),%eax	Move -0x8(rbp) to %eax
	mov -0x4(%rbp),%edx	Move -0x4(rbp) to %edx
	add %edx,%eax	Add values of %edx and %eax
	pop %rbp	Return the output of add
	retq	

File Four: assignment3_4.o		
Functions	**Blocks of Assembly Code**	**Explanation of Functionality**
main	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	sub $0x10,%rsp	
	movl $0x0,-0x8(%rbp)	Assign $0x0 to -0x8(%rbp)
	jmp 0x4007a0 <main+137>	Jump to 0x4007a0
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4006df <DisplayMenu>	Call DisplayMenu function
	mov $0x400886,%edi	Assign $0x400886 to %edi
	callq 0x4004e0 <puts@plt>	Output value
	lea -0x8(%rbp),%rax	Load -0x8(%rbp) to %rax
	mov %rax,%rsi	Move %rax to %rsi
	mov $0x400899,%edi	Assign $0x400899 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x400520 <__isoc99_scanf@plt>	Reads in value
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	cmp $0x3,%eax	Compare $0x3 to %eax
	je 0x40077a <main+99>	Conditional jump to 0x40077a
	mov $0x40089c,%edi	Assign $0x40089c to %edi
	callq 0x4004e0 <puts@plt>	Output value
	lea -0x4(%rbp),%rax	Load -0x4(%rbp) to %rax
	mov %rax,%rsi	Move %rax to %rsi
	mov $0x400899,%edi	Assign $0x400899 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x400520 <__isoc99_scanf@plt>	Reads in value
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	cmp $0x1,%eax	Compare $0x1 to %eax
	jne 0x40078e <main+119>	Conditional jump to 0x400785
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax
	mov %eax,%edi	Move %eax to %edi
	callq 0x40062d <PrintFact>	Call PrintFact function
	jmp 0x4007a0 <main+137>	Jump to 0x4007a0
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	cmp $0x2,%eax	Compare $0x2 to %eax
	jne 0x4007a0 <main+137>	Conditional jump to 0x4007a0
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax
	mov %eax,%edi	Move %eax to %edi
	callq 0x400688 <PrintSum>	Call PrintSum function
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	cmp $0x3,%eax	Compare $0x3 to %eax
	jne 0x400728 <main+17>	Conditional jump to 0x400728
	mov $0x0,%edi	Exits the main function and ends the program
	callq 0x400530 <exit@plt>	
DisplayMenu	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	mov $0x400851,%edi	Assign $0x400851 to %edi

	callq 0x4004e0 <puts@plt>	Output value of %edi
	mov $0x400864,%edi	Assign $0x400864 to %edi
	callq 0x4004e0 <puts@plt>	Output value of %edi
	mov $0x400871,%edi	Assign $0x400871 to %edi
	callq 0x4004e0 <puts@plt>	Output value of %edi
	mov $0x40087e,%edi	Assign $0x40087e to %edi
	callq 0x4004e0 <puts@plt>	Output value of %edi
	mov $0x400851,%edi	Assign $0x400851 to %edi
	callq 0x4004e0 <puts@plt>	Output value of %edi
	pop %rbp	Return to main function
	retq	
PrintFact	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	sub $0x20,%rsp	Subtract $0x20 from %rsp
	mov %edi,-0x14(%rbp)	Move %edi to -0x14(rbp)
	movl $0x1,-0x4(%rbp)	Assign $0x1 to -0x4(%rbp)
	mov -0x14(%rbp),%eax	Move -0x4(%rbp) to %eax
	mov %eax,-0x8(%rbp)	Move %eax to -0x8(%rbp)
	jmp 0x400669 <PrintFact+60>	Jump to 0x400669
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax
	imul -0x8(%rbp),%eax	Multiply -0x8(%rbp) by %eax
	mov %eax,-0x4(%rbp)	Move %eax to -0x4(%rbp)
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	mov %eax,%esi	Move %eax to %esi
	mov $0x400844,%edi	Assign $0x400844 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4004f0 <printf@plt>	Output value
	subl $0x1,-0x8(%rbp)	Subtract 1 from -0x8(%rbp)
	cmpl $0x0,-0x8(%rbp)	Compare $0x0 to -0x8(%rbp)
	jg 0x400647 <PrintFact+26>	Conditional jump to 0x400647
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax
	mov %eax,%esi	Move %eax to %esi
	mov $0x400848,%edi	Assign $0x400848 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4004f0 <printf@plt>	Output value
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax and return to main function
	leaveq	
	retq	
PrintSum	push %rbp	Moves value of %rsp to %rbp
	mov %rsp,%rbp	
	sub $0x20,%rsp	Subtract $0x20 from %rsp
	mov %edi,-0x14(%rbp)	Move %edi to -0x14(%rbp)
	movl $0x0,-0x4(%rbp)	Assign $0x0 to -0x4(%rbp)
	mov -0x14(%rbp),%eax	Move -0x14(%rbp) to %eax
	mov %eax,-0x8(%rbp)	Move %eax to -0x8(%rbp)
	jmp 0x4006c0 <PrintSum+56>	Jump to 0x4006c0
	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	add %eax,-0x4(%rbp)	Add %eax and -0x4(%rbp)

	mov -0x8(%rbp),%eax	Move -0x8(%rbp) to %eax
	mov %eax,%esi	Move %eax to %esi
	mov $0x400844,%edi	Assign $0x400844 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4004f0 <printf@plt>	Output value
	subl $0x1,-0x8(%rbp)	Subtract $0x1 from -0x8(%rbp)
	cmpl $0x0,-0x8(%rbp)	Compare $0x0 to -0x8(%rbp)
	jg 0x4006a2 <PrintSum+26>	Conditional jump to 0x4006a2
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax
	mov %eax,%esi	Move %eax to %esi
	mov $0x400848,%edi	Assign $0x400848 to %edi
	mov $0x0,%eax	Assign $0x0 to %eax
	callq 0x4004f0 <printf@plt>	Output value
	mov -0x4(%rbp),%eax	Move -0x4(%rbp) to %eax and return to main function
	leaveq	
	retq	