

ICS'12 Homework 3

1. Describe the behavior of the instruction: `pushl %esp;`

2. We have the following values in the corresponding memory address and registers.

Address	Values
0x004	0x34
0x00c	0x12
0x104	0x5FA
0x164	0x8
0x900	0x8932FF

Register	Value
%eax	0x004
%ecx	0x100
%edx	0x4
%ebx	0x10

Fill in the blanks with the right values

Operand	Value
%ebx	
0x900	
(%eax)	
8(%eax)	
\$0xA12	
(%eax,%ecx)	
(%eax,%edx,2)	
0x24(%ecx,%ebx,4)	

3 .Fill the blanks according to the values of registers and memory:

Address	Value
0x100	0xCD
0x104	0x46
0x110	0xAA
0x114	0x23

Register	Value
%eax	0x100
%ecx	0xCF
%edx	0x4

Instruction	Destination	Value
<i>addl %ecx, %eax</i>		
<i>sarl \$3, %eax</i>		
<i>subl %edx, (%eax)</i>		
<i>decl 0x10(%eax)</i>		
<i>imull \$2, (%eax, %edx, 4)</i>		
<i>leal 6(\$eax), %ecx</i>		

4 .Here is a prototype of a function:

```
int decode(int x,int y,int z);
```

The IA32 assembly code of this function is:

X at %ebp+8, y at %ebp+12, z at &ebp+16

1. *movl 16(%ebp), %edx*

2. *subl 12(%ebp), %edx*

3. `movl %edx, %eax`
4. `sall $15, %eax`
5. `sarl $15, %eax`
6. `xorl 8(%ebp), %edx`
7. `imull %edx, %eax`

The code stores the return value in the register %eax. Please write the C code according to the assembly code.

5. Suppose we have the following code fragment:

- | | |
|-------------------------------|------------------------|
| <code>pushl %ebp</code> | <i>(Instruction 1)</i> |
| <code>movl %esp, %ebp</code> | <i>(Instruction 2)</i> |
| <code>movsbl %dl, %ecx</code> | <i>(Instruction 3)</i> |
| <code>movzbl %ch, %eax</code> | <i>(Instruction 4)</i> |
| <code>movl %ebp, %esp</code> | <i>(Instruction 5)</i> |
| <code>popl %ebp</code> | <i>(Instruction 6)</i> |

Before the execution of the first instruction of the above code, the register values are:

`%edx = 0x55AA`

`%esp = 0xBFFFFFFA0`

`%ebp = 0xBFFFFFFC4`

Fill in the blanks with the right values:

After instruction 2 has finished:

<i>%edx</i>	
<i>%esp</i>	
<i>%ebp</i>	

After instruction 3 has finished:

<i>%edx</i>	
<i>%esp</i>	
<i>%ecx</i>	

After instruction 4 has finished:

<i>%edx</i>	
<i>%ecx</i>	
<i>%eax</i>	

After instruction 6 has finished:

<i>%eax</i>	
<i>%esp</i>	
<i>%ebp</i>	