

1.Addressing:

We have following value address binding table:

Address	Value	Register	Value
0x002	0x105	%eax	0x02
0x310	0x106	%ebx	0x10
0x334	0x107	%ecx	0x300
0x316	0x108	%edx	0x50
0x322	0x109		

Operand	Value
%eax	0x02
(%eax)	0x105
\$0x310	0x310
(%ebx,%ecx)	0x106
16(%ecx)	0x106
0x002	0x105
0x30(%ecx,%eax,2)	0x107

2. Describe the behavior of the instruction : pushl %esp;

$M[R[ESP] - 4] = R[ESP]$

$R[ESP] = R[ESP] - 4;$

3.Suppose we have the following code fragment:

Instruction 1 : pushl %ebp
Instruction 2 : movl %esp,%ebp
Instruction 3 : movsbl %al,%ebx
Instruction 4 : movzbl %bh,%ecx
Instruction 5 : pushl %esp
Instruction 6 : movl %ebp,%esp
Instruction 7 : popl %ebp

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

%eax = 0x7788

%esp = 0xBFFFFFFA0

%ebp = 0xBFFFFFFB0

After instruction 1 finished :

%ebp	0xBFFFFFFB0
%esp	0xBFFFFFF9C
%eax	0x7788

After instruction 2 finished :

%ebp	0xBFFFFFF9C
%esp	0xBFFFFFF9C
%eax	0x7788

After instruction 3 finished :

%esp	0xBFFFFFF9C
%ebx	0xFFFFFFFF88
%eax	0x7788

After instruction 4 finished :

%eax	0x7788
%ebx	0xFFFFFFFF88
%ecx	0xFF

After instruction 5 finished :

%ebp	0xBFFFFFF9C
%esp	0xBFFFFFF98
(%esp)	0xBFFFFFF9C

After instruction 6 finished :

%ebp	0xBFFFFFF9C
%esp	0xBFFFFFF9C
%eax	0x7788

After instruction 7 finished :

%ebp	0xBFFFFFFB0
%esp	0xBFFFFFFA0
%eax	0x7788

4.

```
Int a [10];  
Void f(void){  
    Int i ;  
    For(i = 0; i<10;i++)  
        a[i] = i;  
}
```

The address of a is 0X100 (in fact you can not access the memory address 0x100)

Q1:after the function f () finished , fill the following table(if you do not know the value input NONE)

%ecx	0x100
%eax	5

\$0x100	0x100
0x104	0x1
0x0FC	NONE
(%ecx,%eax,4)	5