HW 6

1. Jump Table.

Consider the assembly dump for a switch and jump table given below. The switch expression and all case and break statements have been removed from the C code below it. Using what you know, fill in the switch expression, as well as case statements and break statements on the lines below, noting that you may not use every line.

```
void scramble(unsigned a, int * b, int * c)
{
   switch(___) {
   _____________________
   c[5] = *b;
   _____________________
   _____________________
   _____________________
   *b += *c;
   _____________________
   _____________________
   *b = *c;
   _____________________
   _____________________
   *b *= 3;
   _____________________
   _____________________
   *b = 15213;
   _____________________
   }
}
```

```
L1:
  .quad L6
  .quad L4
  .quad L5
  .quad L3
  .quad L4
  .quad L4
  .quad L2
  .quad L6

scramble:
  cmpq $7, %rdi
  ja L4
  jmp *L1(,%rdi, 8)
L2:
  movl (%rdx), %r8d
  addl %r8d, (%rsi)
  jmp L7
L3:
  movl (%rsi), %r8d
  mov $5, %r9
  movl %r8d, (%rdx, %r9, 4)
  jmp L7
L4:
  movl $15213, (%rsi)
  jmp L7
L5:
  movl (%rdx), %r8d
  movl %r8d, (%rsi)
L6:
  movl (%rsi), %r8d
  lea (%r8, %r8, 2), %r8
  movl %r8d, (%rsi)
  jmp L7
L7:
  ret
```

2. Translation

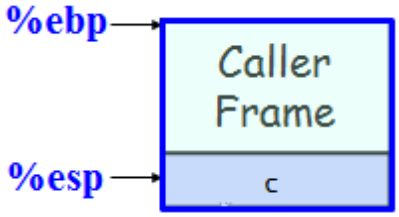
Translate the following assembly into C codes. Note that you can name local variables represented by 28(%esp), 24(%esp)… freely as you like.

```
movl $1, 28(%esp)
  movl $1, 24(%esp)
  movl $10, 16(%esp)
  movl $3, 20(%esp)
  jmp L2
L3:
  movl 28(%esp), %eax
  movl %eax, 12(%esp)
  movl 24(%esp), %eax
  movl %eax, 28(%esp)
  movl 12(%esp), %eax
  addl %eax, 24(%esp)
  incl 20(%esp)
L2:
  cmpl $10, 20(%esp)
  jle L3
  (Following code omitted…)
```

3. Function Call

Draw a stack for each step when the following function is being calledand indicate where %ebp and %esp points. Here is an example where the first step takes place.

```
int foo(int a, int b, int c);
```



4. Switch

Translate the following switch statements into assembly using jump table.

```
int x = <some value>;
int result = 0;
switch (x) {
   case 64:
      result = x + x;
      break;
   case 66:
      result = x + 10;
      break;
   case 68:
      result = x * 2;
      // Notice: there is no break
      here!
   case 69:
      result = result + 1;
      break;
   default:
      result = 3;
      break;
   }
```