1. **x86:**

The following is x86 assembly. Write equivalent code in C and MIPS assembly. (It will probably be easier for you to write the C code first, and then the MIPS assembly equivalent.) In your write-up, include a mapping between the C variables / MIPS registers you use and the x86 registers.

```assembly
.text
.globl main
main:
    movl $0x0, %ebx
    movl $0x0, %ecx
    jmp check
loop:
    mov %ecx, %eax
    add %eax, %ebx
    addl $0x1, %ecx
check:
    cmpl $0x63, %ecx
    jle loop
    ret
```

**Ans:**

**C Code:**

```c
void main(void) {
    int counter; // %ecx
    int base = 0; // %ebx
    int acc; // %eax
    for (counter = 0; counter < 100; counter++) {
```
acc = counter;
    base += acc;
}
2. **VLIW:**

In this problem, you are given a tool to transform machine code that is written for the MIPS ISA to code in a VLIW ISA. The VLIW ISA is identical to MIPS except that multiple instructions can be grouped together into one VLIW instruction. Up to N MIPS instructions can be grouped together (N is the machine width, which depends on the particular machine). The transformation tool cannot reorder MIPS instructions to fill VLIW instructions. You are given the following MIPS program (we have numbered the instructions for reference below):

(01) `lw $t0, 0($a0)`
(02) `lw $t2, 8($a0)`
(03) `lw $t1, 4($a0)`
(04) `add $t6, $t0, $t1`
(05) `lw $t3, 12($a0)`
(06) `sub $t7, $t1, $t2`
(07) `lw $t4, 16($a0)`
(08) `lw $t5, 20($a0)`
(09) `srlv $s2, $t6, $t7`
(10) `sub $s1, $t4, $t5`
(11) `add $s0, $t3, $t4`
(12) `sllv $s4, $t7, $s1`
(13) `srlv $s3, $t6, $s0`
(14) `sllv $s5, $s0, $s1`
(15) `add $s6, $s3, $s4`
(16) `add $s7, $s4, $s6`
(17) `srlv $t0, $s6, $s7`
(18) `srlv $t1, $t0, $s7`
1. When you run the tool with its settings targeted for a particular VLIW machine, you find that the resulting VLIW code has 9 VLIW instructions. What minimum value of N must the target VLIW machine have?

Ans:

N = 3 (see VLIW program below). If N = 2, then the VLIW program must have at least 11 MIPS instructions, and the number of VLIW instructions either stays the same or decreases as width is increased by one MIPS instruction.

2. Write the MIPS instruction numbers (from the code above) corresponding to each VLIW instruction, for this value of N.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>02</td>
<td>03</td>
</tr>
<tr>
<td>2</td>
<td>04</td>
<td>05</td>
<td>06</td>
</tr>
<tr>
<td>3</td>
<td>07</td>
<td>08</td>
<td>09</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Multiple answers are accepted.