

1.

```

80000000 <main>:
80000000:      00100413          li      s0,1
80000004:      00000493          li      s1,0
80000008:      04000293          li      t0,64

8000000c <loop>:
8000000c:      00540863          beq     s0,t0,8000001c <salto>
80000010:      00141413          slli   s0,s0,0x1
80000014:      00148493          addi   s1,s1,1
80000018:      ff5ff06f          j      8000000c <loop>

8000001c <salto>:
8000001c:      00000013          nop
    
```

2.

A operação é uma sequência de shift lógico para a esquerda imediato do valor do registrador s0 com 1 que tem seu valor inicial 1, que é o mesmo que multiplicar por 2, até que ele chegue ao valor de t0 que é 64

3.

```

x0  zero: 00  ra : 00  sp : 00  gp : 00
x4  tp   : 00  t0 : 40  t1 : 00  t2 : 00
x8  s0   : 40  s1 : 06  a0 : 00  a1 : 00
x12 a2   : 00  a3 : 00  a4 : 00  a5 : 00
    
```

4. o valor do Loop é 0C pois ele é a quarta instrução executada pela máquina e o valor do salto é 1C pois é a oitava instrução caso o valor de s0 seja igual ao de t0 a execução é direcionada para a instrução de salto caso contrário ele segue a ordem de instruções e ao chegar na instrução 18 ele pula de volta para a instrução 0C

```

80000000 <main>:
80000000:      00100413          li      s0,1
80000004:      00000493          li      s1,0
80000008:      04000293          li      t0,64

8000000c <loop>:
8000000c:      00540863          beq     s0,t0,8000001c <salto>
80000010:      00141413          slli   s0,s0,0x1
80000014:      00148493          addi   s1,s1,1
80000018:      ff5ff06f          j      8000000c <loop>

8000001c <salto>:
8000001c:      00000013          nop
    
```