```
1   int dw_loop(int x, int y, int n) {
2       do {
3           x += n;
4           y *= n;
5           n--;
6       } while ((n > 0) && (y < n));
7       return x;
8   }
```

GCC menghasilkan kode assembly berikut:

```
    x at %ebp+8, y at %ebp+12, n at %ebp+16
1     movl    8(%ebp), %eax
2     movl    12(%ebp), %ecx
3     movl    16(%ebp), %edx
4   .L2:
5     addl    %edx, %eax
6     imull   %edx, %ecx
7     subl    $1, %edx
8     testl   %edx, %edx
9     jle     .L5
10    cmpl    %edx, %ecx
11    jl      .L2
12  .L5:
```

a. Buatlah tabel penggunaan register, yg berisi nama register, variable yang disimpan pada register, dan nilai awal variabel.

menyimpan variable x

menyimpan variable y

menyimpan variable n

n ≤ 0 → keluar dan loop

n

y

y < n

testl → membandingkan dengan 0

jle → lebih kecil sama dengan

jl → lebih kecil dari

2. Sebuah fungsi fun_a memiliki struktur berikut:

```
int fun_a(unsigned x) {
    int val = 0;
    while ( X != 0 ) {
        val ^= x;
        x >>= 1
    }
    return val&=1;
}
```

kalo x=0 dimakan keluar

xor

Hasil kompilasi GCC adalah sbb:

```
    x at %ebp+8
1     movl    8(%ebp), %edx
2     movl    $0, %eax
3     testl   %edx, %edx
4     je      .L7
5   .L10:
6     xorl    %edx, %eax
7     shrl    %edx
8     jne     .L10
9   .L7:
10    andl    $1, %eax
```

nyimpen x

nyimpen val : 0

kasus kalo dia ≠ 0

Shift right by 1

Isilah bagian kosong pada kode C di atas dengan kode sesuai assembly nya.

3. Fungsi fun_b memiliki struktur berikut:

```
int fun_b(unsigned x) {
    int val = 0;
    int i;
    for ( i = 0 ; i != 32 ; i++ ) {
        val = 2 * val || x & 1 ;
    }     x = x >> 1 ;
    return val;
}
```

Dengan hasil kompilasi gcc sbb:

```
    x at %ebp+8
1       movl    8(%ebp), %ebx
2       movl    $0, %eax  -> val
3       movl    $0, %ecx
4   .L13:

5       leal    (%eax,%eax), %edx   edx : 2 * val
6       movl    %ebx, %eax
7       andl    $1, %eax            val : x & 1
8       orl     %edx, %eax          val : 2 * val || (x & 1)
9       shrl    %ebx    x >>= 1     Shift right by 1
10      addl    $1, %ecx    i ++
11      cmpl    $32, %ecx   i != 32
12      jne     .L13
```