1.

```
Void doit(void *args){
    Int *num = (int*)args;
    For(int i = 0; i < M; i++){
        Hash_insert(hashtb, num[i]);
}
```

```
2. void init(counter *c){
    c->value = 0;
}
void increment(counter_t *c){
    fetch_and_add(&(c->value), 1);
}
void decrement(counter_t *c){
    fetch_and_add(&(c->value), -1);
}
int get(counter_t *c){
    return c->value;
}
```

3. a 的初值为 1，b 的初值为 0

```
void *t1(void *arg){
    [1]: P(&a);
    [2]: V(&b);    V(&b);
}
void *t2(void *arg){
    [1]: P(&b);
    [2]:if (b == 0) V(&a);
}
```

4.

```
Int CAS(unsigned long *dst, unsigned long oldVal, unsigned long newVal){
    Unsigned char ret;
    __asm__ __volatile__(
        “ lock: cmpxchgq %2,%1\n”
        “ sete %0\n”
        : “=q”(ret), “=m”(*dst)
        :”r”(newVal), “m”(*dst),  “a”(oldVal)
        :”memory”);
    If(ret) return 1;
    Return 0;
}
Void List_Insert(){
    Node_t *new_node = malloc(sizeof(node_t));
```

```
        If(new_node == NULL){
                perror(“malloc”);
                return;
        }
        new_node->key = key;
        int tmp;
        do{
                new_node ->next = L->head;
                tmp = CAS(&(L->head), new_node->next, new_node);
        }while(tmp == 0);
}
```