

(یعنی اجزای ۲ و ۳ نیز تمام شود)

الف) بله، ممکن است. اگر در هسته اول $f_{001}(1)$ کامل اجرا شود سپس هسته دوم $f_{001}(1)$ را اجرا کند و قبل از حد شدن f_{002} ، هسته اول $f_{003}(1)$ را اجرا کند، هسته اول در تابع f می‌بایستد و نمی‌تواند وارد f شود و همچنین هسته دوم در f بوده و نمی‌تواند وارد f شود و در این حالت $head \rightarrow next$ به وجود می‌آید.

(ب)

۱) به تعداد اعداد موجود در لیست، تسک ساخته می‌شود.
۲) اندازه تسک‌ها می‌تواند بسیار متفاوت باشد، برای مثال اگر همه‌ی اعداد 1 یا 1000 باشند، به تعداد اعضای لیست تسک ساخته می‌شود ولی تمام تسک‌های 1 بسیار کوتاه‌تر بوده و صرفاً باعث اتلاف منابع می‌شود.

(ج)

```
#pragma omp parallel
{
    #pragma omp single nowait
    {
        int len = 0;
        int sum = 0;
        node *p = head;
        while (p) {
            sum += p->value;
            len++;
            p = p->next;
        }
        int av = sum / len;
        p = head;
        node *ls = head;
        int cur = 0;
        while (p)
            cur += p->value;
        if (cur > av || !p->next)
            // ...
    }
}
```

```
#pragma omp taskwait
while (last != p) {
    Print is Print(last->value);
    last = last->next;
}
last = p
p = p->next
}
```