

Листинг программы:

```
int n;
std::cin >> n;

int k;
std::cin >> k;

int sum = 0;
#pragma omp parallel for reduction(+ \
                                   : sum)
num_threads(k)
for (int i = 0; i < n; i++)
{
    sum += i;
    printf("[%d]: Sum = %d\n", omp_get_thread_num(),
sum);
}

std::cout << sum;

return 0;
```

Ответы на вопросы:

1) Как оформляются в программе результаты параллельного выполнения потоков?

Оформление результатов параллельного выполнения потоков зависит от конкретной задачи и структуры программы. Одним из способов является использование общей переменной для сохранения результатов, на которую каждый поток добавляет свой вклад. Для того чтобы избежать конфликта доступа к общей переменной со стороны нескольких потоков, можно использовать директиву "critical" или "atomic". Другим способом является использование массивов, где каждый поток записывает свои результаты в отдельный элемент массива.

2) Какие дополнительные действия выполняет директива, если она имеет параметр reduction?

Директива с параметром reduction используется для выполнения операций сокращения над переменными, такими как сумма, произведение, минимум, максимум и т.д. Она выполняет дополнительные действия, такие как объявление приватной переменной для каждого потока, инициализация ее начальным значением и автоматическое сокращение результатов каждого потока в одно окончательное значение.

3) Чем инициализируются частные переменные, создаваемые параметром reduction?

Частные переменные, создаваемые параметром reduction, инициализируются в соответствии с операцией сокращения, определенной в директиве. Например, если операция сокращения - это сумма, то переменная будет инициализироваться нулем, а если операция сокращения - это минимум, то переменная будет инициализироваться максимальным значением типа переменной. Если начальное значение переменной задается явно в директиве, то оно используется в качестве начального значения для всех приватных переменных.