سوال اول :

کد اول : مشکل اصلی این کد race condition است. برای حل این مشکل دو کد زیر را می توانیم استفاده کنیم :

```
#pragma omp parallel for num_threads(n_t) reduction(+:acc)
	for (int i = 0; i < 100; i++) {
		acc++;
	}
```

یا:

```
#pragma omp parallel for private(i)
	for (i = 0; i < 100; i++) {
		#pragma omp atomic
		acc++;
```

برای محاسبه تسریع از فرمول زیر استفاده می کنیم:

$$speed\ up = \frac{T_s}{T_p}$$

برای محاسبه تسریع هر الگوریتم به دلیل تصادفی بودن اعداد ، آزمایش را 10 بار تکرار کرده و از تسریع های بدست آمده میانگین می گیریم

برای این الگوریتم داریم :

$$speed\ up = 0.000109$$

دلیل پایین شدن تسریع کوچک بودن مساله است که اجرای سریال ، به دلیل سربار اضافه اجرای موازی، برای ما بهتر است.

البته داخل کد داده شده خطای syntax نیز داشته ایم و #pragme omp for حتما باید داخل بلاک parallel بیاید.

کد دوم :

مشکل اصلی این کد locality of refrence است ، در واقع یک آرایه دو بعدی در حافظه RAM به صورت زیر ذخیره می شود:

A[0][0] A[0][4] A[0][5] ... A[1][0] A[1][6] A[1][7] ... A[2][0] A[2][8] A[2][9] ..

بنابراین اگر بخواهیم این آرایه دو بعدی را مقدار دهی کنیم بهتر است به صورت سطری آن را پیمایش کنیم:

```
unsigned int n_t = 4;
int E[300][300];
double start = omp_get_wtime();
#pragma omp parallel for num_threads(n_t)
for (int i = 0; i < 250; i++)
       for (int j = 0; j < 250; j++)
                       E[i][j] += j;
double end = omp_get_wtime() - start;
```

این پیمایش نیز همان نتیجه مطلوب را به ما می دهد.

$$speed\ up = 0.002672$$

کد سوم:

در کد داده شده بهتر است خط زیر را :

```
int workload_size = arr_size / 4;
```

به خط زیر تغییر دهیم :

```
int nthd = omp_get_num_threads();
int workload_size = arr_size / nthd;
```

زیرا ممکن است 4 نخ در حال اجرا نباشند.

اما یک بهبود بهتر به صورت زیر است :

```
#pragma omp parallel for
for (int i = 0; i < arr_size; i++) {
	arr[i] = 0;
}
```

$$speed\ up = 0.052674$$

البته مشکل دیگری که این کد داشت این است که خانه های آخر آن مقدار دهی نمی شوند و متغییر E تعریف شده و از آن استفاده نشده است.

سوال دوم :

کد مربوط به موازی سازی بلاکی :

```
void matmul3d_block_p(Tensor3D* A, Tensor3D* B, Tensor3D* C) {
    #pragma omp parallel for
    for (int i = 0; i < MATRIX_SIZE; i++) {
        for (int j = 0; j < MATRIX_SIZE; j++) {
            for (int k = 0; k < MATRIX_SIZE; k++) {
                for (int p = 0; p < MATRIX_SIZE; p++) {
                    C->data[i][j][k] += A->data[i][j][p] * B->data[i][p][k];
                }
            }
        }
    }
}
```

در این کد ضرب هر دو ماتریس دو بعدی متناظر در دو ماتریس سه بعدی به یک تسک تبدیل می شود.

موازی سازی سطری :

```
void matmul3d_row_p(Tensor3D* A, Tensor3D* B, Tensor3D* C) {
    #pragma omp parallel for collapse(2)
    for (int i = 0; i < MATRIX_SIZE; i++) {
        for (int j = 0; j < MATRIX_SIZE; j++) {
            for (int k = 0; k < MATRIX_SIZE; k++) {
                for (int p = 0; p < MATRIX_SIZE; p++) {
                    C->data[i][j][k] += A->data[i][j][p] * B->data[i][p][k];
                }
            }
        }
    }
}
```

موازی سازی ستونی :

```
void matmul3d_col_p(Tensor3D* A, Tensor3D* B, Tensor3D* C) {
    #pragma omp parallel for collapse(3)
    for (int i = 0; i < MATRIX_SIZE; i++) {
        for (int j = 0; j < MATRIX_SIZE; j++) {
            for (int k = 0; k < MATRIX_SIZE; k++) {
                for (int p = 0; p < MATRIX_SIZE; p++) {
                    C->data[i][j][k] += A->data[i][j][p] * B->data[i][p][k];
                }
            }
        }
    }
}
```

`Collapse` برای تقسیم `iteration` های `nested loop` ها و تخصیص آن ها به نخ ها استفاده می شود.

جداول مربوط به هر روش (به دلیل طولانی بودن محاسبات مربوط به ماتریس با ابعاد `1024` ، از اندازه گیری زمان آن خودداری شده است) :

Block

تعداد نخ	۱۲۸	۲۵۶	۵۱۲	تسریع
16	0.3121	4.737004	111.299558	3.8817
8	0.2995	4.784186	104.640169	4.1288
4	0.3257	5.162921	122.550877	3.5253
1	1.1322	18.177092	432.038447	1

row

تعداد نخ	۱۲۸	۲۵۶	۵۱۲	تسریع
16	0.313805	4.943173	105.353384	4.1509
8	0.297501	4.962854	106.723253	4.0976
4	0.308830	5.192666	123.453338	3.5423
1	1.138118	18.591885	437.317068	1

col

تعداد نخ	۱۲۸	۲۵۶	۵۱۲	تسریع
16	0.318444	4.911952	105.853580	4.1952
8	0.303149	4.934513	105.338830	4.2157
4	0.310401	5.255514	120.716820	3.6787
1	1.134222	18.591111	444.081872	1