

Soru 7: Soru 1 için 2 boyutlu matrisimizi belli bir oranda scale etmemiz gerekiyor. Bir önceki ödevde algoritmasını oluşturmuştum. kısaca özet geçip kodunu yazacağım.



A matrisindeki veriyi b matrisinde duplicate ederek yerleştirmeliyiz. B matrisinde 2for ile gezerken yerimizi  $i, j$  - scale factor kadar bölersek A matrisindeki yerimize ulaşabiliriz.

Kod:

```
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <omp.h> → kütüphanemizi ekliyoruz

#define N_ROW 600
#define N_COL 800
#define N_SCALE 2

int a[N_ROW][N_COL];
int b[N_ROW*N_SCALE][N_COL*N_SCALE];

int main() {
    struct timeval t;
    double vstart, vend; → Zamanı hesabi için gereken değişkenleri tanımlıyoruz

    gettimeofday(&t, NULL);
    vstart = t.tv_sec + t.tv_usec * 1.e-6; → Başlangıç zamanımızı alıyoruz

    #pragma omp parallel for → forun paralel olacağını belirtiyoruz
    for (int row=0; row<N_ROW*N_SCALE; row+=N_SCALE) {
        for (int col=0; col<N_COL*N_SCALE; col+=N_SCALE) {
            int data = a[row/N_SCALE][col/N_SCALE];
            b[row][col] = data;
            b[row][col+1] = data;
            b[row+1][col] = data;
            b[row+1][col+1] = data;
        }
    }

    gettimeofday(&t, NULL);
    vend = t.tv_sec + t.tv_usec * 1.e-6; → Bitiş zamanını alıyoruz
    printf("elapsed time: %d", vend-vstart); → ekrana basıyoruz
    return 0;
}
```

Furkan Kaya 191216002 fk

Soru 2 : Bu soruda ise A ile B matrislerini çarpıp c matrisine atanmalıyım. Yine bir önceki ödevde bulunduğum algoritmayı koddım ve openmp ile paralel hale getirmeliyim.

Kod :

```
#include <stdio.h> #include <time.h> #include <sys/time.h>
```

```
#include <omp.h> → kütüphane mizi ekliyoruz
```

```
#define N 1000
```

```
int a[N][N];
```

```
int b[N][N];
```

```
int c[N][N];
```

```
int main() {
```

```
    struct timeval t;
```

```
    double vstart, vend;
```

→ zaman ölçümü için gerekli değişkenleri tanımlıyoruz

```
    gettimeofday(&t, NULL);
```

```
    vstart = t.tv_sec + t.tv_usec * 1.e-6;
```

→ başlangıç zamanımızı alıyoruz

```
    #pragma omp parallel for
```

```
    for (int row = 0; row < N; row++) {
```

```
        for (int col = 0; col < N; col++) {
```

```
            int sum = 0;
```

```
            for (int walker = 0; walker < N; walker++) {
```

```
                sum += a[row][walker] * b[walker][col];
```

```
            }
```

```
            c[row][col] = sum;
```

```
        }
```

```
    }
```

```
    gettimeofday(&t, NULL);
```

```
    vend = t.tv_sec + t.tv_usec * 1.e-6;
```

→ bitiş zamanımızı alıyoruz

```
    printf("time: %d", vend - vstart);
```

```
    return 0;
```

3