Questão 01 (0,5 pontos)

Considere o código a seguir para multiplicação de matrizes:

```
double **mtxMul (double **c, double **a, double **b, int n) {
   for (int i = 0; i < n; i++)
       for (int j = 0; j < n; j++) {
            c[i][j] = 0.0;
            for (int k = 0; k < n; k++)
                c[i][j] = c[i][j] + a[i][k] * b[k][j];
        }
   return c;
}
```

Suponha que duas matrizes 100 x 100 são multiplicadas usando 8 threads. Quantos produtos escalares, isto é, operações realizadas pelo laço for mais interno, cada thread deve calcular se as seguintes abordagens para paralelizar os dois laços for mais externos forem utilizadas:

a) Apenas o laço for mais externo é paralelizado.

O valor de iteração 100 é dividido para 8 threads, ou seja, 4 threads ficam com 12 valores e 4 threads com 13.
Cada iteração realizará 100 produtos escalares, então as 4 threads com 12 valores, realizará 12*100 = 1200 produtos escalares, já as threads com 13 valores irão realizar 13*100 = 1300 produtos escalares. Totalizando 1200*4+1300*4 = 10000 produtos.

b) A diretiva collapse(2) é aplicada ao laço mais externo.

A quantidade de iterações no total será 100*100 = 10000, ou seja o valor total de produtos escalares. Esse valor será igualmente distribuído nas 8 threads 10000/8 = 1250 produtos escalares por thread.

c) Ativando regiões paralelas aninhadas e OMP_NUM_THREADS=2,4.A

O primeiro FOR irá ser dividido em duas threads, ou seja, cada thread terá 50 valores na iteração, e cada thread terá o iterador do segundo FOR dividido para 4 threads na segunda região paralela, então cada thread terá 50 valores do primeiro FOR e 25 do segundo FOR, totalizando 50*25 = 1250 produtos escalares por thread.

Questão 02 (0,5 pontos)

Considere o programa abaixo:

```
#include <stdio.h>
#include <unistd.h>
```

```
#include <omp.h>

int main (int argc , char *argv[]) {
    int max;
    sscanf (argv[1], "%d", &max);
    long int sum = 0;
    #pragma omp parallel for reduction(+:sum) schedule(runtime)
    for (int i = 1; i <= max; i++) {
        printf ("%2d @ %d\n", i, omp_get_thread_num());
        sleep (i < 4 ? i + 1 : 1);
        sum = sum + i;
     }
    printf ("%ld\n", sum);
    return 0;
}
```

Suponha que o programa está em execução com 4 threads e o valor fornecido para max foi 20. Determine qual iteração será executada por qual thread se as seguintes diretivas forem utilizadas como estratégias de escalonamento.

a) static,1, static,2 e static,3

static 1:

thread 1: 1, 5, 9, 13, 17
thread 2: 2, 6, 10, 14, 18
thread 3: 3, 7, 11, 15, 19
thread 4: 4, 8, 12, 16, 20

Thread 1	1	1	5	9	13	17			
Thread 2	2	2	2	6	10	14	18		
Thread 3	3	3	3	3	7	11	15	19	
Thread 4	4	8	12	16	20				

static 2:

thread 1: 1, 2, 9, 10, 17, 18
thread 2: 3, 4, 11, 12, 19, 20
thread 3: 5, 6, 13, 14
thread 4: 7, 8, 15, 16

Thread 1	1	1	2	2	2	9	10	17	18
Thread 2	3	3	3	3	4	11	12	19	20
Thread 3	5	6	13	14					
Thread 4	7	8	15	16					

static 3:

thread 1: 1, 2, 3, 13, 14, 15
thread 2: 4, 5, 6, 16, 17, 18
thread 3: 7, 8, 9, 19, 20
thread 4: 10, 11, 12

Thread 1	1	1	2	2	2	3	3	3	3
Thread 2	4	5	6	16	17	18			
Thread 3	7	8	9	19	20				
Thread 4	10	11	12						

b) dynamic,1, dynamic,2 e dynamic,3

dynamic 1:

thread 1: 1, 6, 10, 14, 18
thread 2: 2, 8, 13, 17
thread 3: 3, 11, 15, 19
thread 4: 4, 5, 7, 9, 12, 16, 20

Thread 1	1	1	6	10	14	18			
Thread 2	2	2	2	8	13	17			
Thread 3	3	3	3	3	11	15	19		
Thread 4	4	5	7	9	12	16	20		

dynamic 2:

thread 1: 1, 2, 17,18
thread 2: 3, 4, 19, 20
thread 3: 5, 6, 9, 10, 13, 14
thread 4: 7, 8, 11, 12, 15, 16

Thread 1	1	1	2	2	2	17	18		
Thread 2	3	3	3	3	4	19	20		
Thread 3	5	6	9	10	13	14			
Thread 4	7	8	11	12	15	16			

dynamic 3:

thread 1: 1, 2, 3
thread 2: 4, 5, 6, 13, 14, 15
thread 3: 7, 8, 9, 16, 17, 18
thread 4: 10, 11, 12, 19, 20

Thread 1	1	1	2	2	2	3	3	3	3
Thread 2	4	5	6	13	14	15			
Thread 3	7	8	9	16	17	18			
Thread 4	10	11	12	19	20				