Guliane Gomes Fernandes Silva 84183

I –

a) $speedup = \frac{1}{1-p} = \frac{1}{1-0,7} = 3,33$

b) $speedup = \frac{1}{\frac{p}{N} + s} = \frac{1}{\frac{0,7}{20} + 0,3} = 3,985$

II –

a)
```
vmax = v[0];
#pragma omp parallel for private(i) shared(v, vmax)
for (i = 1; i < N; i++)
    #pragma omp critical
    if (v[i] > vmax)
        vmax = v[i];
```

vmax não deve ser pantilhada

Então vmax não pode ser pantilhada entre threads, devendo ser firstprivate. Além disso, ~~não~~ #pragma omp critical deveria vir no fim do pragrema para atualizar uma variável global, definida previamente, depois de calculadas as variáveis locais "vmax" de todas as threads.

b)
```
vmax = v[0];
#pragma omp parallel for reduction (max: vmax)
for (i = 1; i < N; i++)
    if (v[i] > max) vmax = v[i];
```

Uma cópia local de variável vmax é feita e inicializada com o valor correspondente à "op" max. Essa variável é atualizada na cópia local. Então cada thread vai ter um vmax final que depois é comparado com o valor de vmax local de todas as outras threads e com o valor inicial vmax = v[0], sendo escolhido o valor máximo dessas todas.

III –

a) paddw → adicione individualmente as words usando non-saturating addition ou seja, faz 16 bits

	BBDO	FF00	5618	a574
+	AAOB	5106	A098	FB1F
	C5BB	5006	F710	A093

b) Paddusbs → adicione individualmente 8 operadores de 8-bits usando signed saturation arithmetic

	BB	00	FF	00	56	78	A5	74
+	0A	0B	01	06	A0	98	FB	1F
	C5	BB	FF	06	F6	FF	FF	93

IV

a) Podemos utilizar, por exemplo, um bloco fixo de 16×16 e uma grelha de tamanho variável.

Seja $h = 256$ e $W = 128$.

Então podemos ter

$A = W/16 = 8 = gridDim.x$ e

$B = h/16 = 16 = gridDim.y$

$C = 16 = blockDim.x$

$D = 16 = blockDim.y$

b) Posto isto, as coordenadas podem ser determinadas da seguinte forma:

$$row = \underbrace{blockIdx.y}_{0-15} * \underbrace{blockDim.y}_{16} + \underbrace{threadIdx.y}_{0-15} \quad \begin{matrix} \rightarrow min = 0 \\ \rightarrow max = 255 \\ h \end{matrix}$$

$$col = \underbrace{blockIdx.x}_{0-7} + \underbrace{blockDim.x}_{16} + \underbrace{threadIdx.x}_{0-15} \quad \begin{matrix} \rightarrow min = 0 \\ \rightarrow max = 127 \\ w \end{matrix}$$

c) Global – é do tipo DRAM e é partilhada por todas as threads de todos os blocos. É de leitura e escrita, exceto se for "texture" ou "constant", pois aí é só de leitura.

Shared – Memória partilhada por todas as threads do mesmo bloco.

Local – Memória acessível só pela própria thread.