## HW8.2. Parallelism Principles

The Hamming distance between two bitstrings of equal length is the number of locations in which the bits differ. For example, hamming(0b1011101, 0b1001001) == 2.

Consider the following implementation of Hamming:

```
uint32_t hamming(uint32_t x, uint32_t y) {
    uint32_t mask, ham_dist = 0;
    #pragma omp for
    for (int i = 0; i <= 31; i++) {
        mask = 1 << i;
        if ((y & mask) != (x & mask))
            ham_dist++;
    }
    return ham_dist;
}</pre>
```

Q1: For each variable, determine if it can be made shared, or must be private in order for the above code to work properly. Note that we generally prefer to use shared variables when possible.

Q1.1: x

- shared
- private

**100%** 

Q1.2: y

- shared
- private

**~** 100%

Q1.3: mask

- shared
- private

**~** 100%

Q1.4: i

- shared
- private

**100%** 

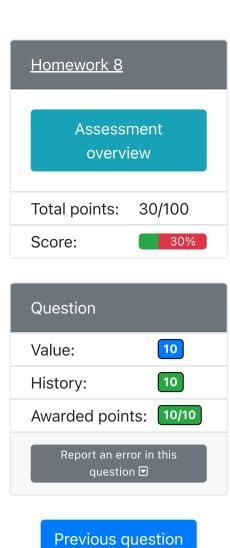
Q1.5: Is a data race possible on any variable that must be shared? If so, what variables cause the data race?

- □ (a) x
- (b) y
- (c) mask
- (d) i
- (e) ham\_dist
- (f) A data race is not possible

Select all possible options that apply. ?

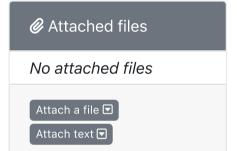
**100%** 

Try a new variant



Tevious question

Next question



Q1.1: x shared Q1.2: y shared Q1.3: mask private Q1.4: i private Q1.1-Q1.4: Note that we only read x and y, so it is safe to share them. For mask, i, the value of the variable needs to be different for each thread, or else all threads would do the same thing. As such, we need to make them private. Q1.5: Is a data race possible on any variable that must be shared? If so, what variables cause the data race? (e) ham\_dist Q1.5: Currently, the addition to ham\_dist is a data race. Two threads can read it at the same time, increment based on their own calculations, and store it back. This would result in incorrect values, since one did not read the updated result from the other. We would either need a critical segment, or a reduction keyword to ensure that ham\_dist is correctly updated. Submitted answer 2 | correct: 100% hide ^ Submitted at 2022-11-09 03:24:05 (PST) Q1: For each variable, determine if it can be made shared, or must be private in order for the above code to work properly. Note that we generally prefer to use shared variables when possible. Q1.1: x shared < 100% Q1.2: y shared 100% Q1.3: mask private 100% Q1.4: i private 100% Q1.5: Is a data race possible on any variable that must be shared? If so, what variables cause the data race? (e) ham\_dist **100%** Submitted answer 1 partially correct: 80% 1 show 🗸 Submitted at 2022-11-09 03:23:50 (PST)

Q1: For each variable, determine if it can be made shared, or must be private in order for the

above code to work properly. Note that we generally prefer to use shared variables when

possible.