

! This quiz has been re-graded; your score was affected.

Quiz-2 Sep 16

Due 17 Sep at 23:55

Points 20

Questions 8

Available 16 Sep at 10:00 - 19 Sep at 23:55 4 days

Time limit 65 Minutes

Instructions

This quiz consists of 8 questions, which are MCQ (multiple choice) or MAQ (multiple answer) questions. It has 20 points. The quiz is open from 10:00 am Sep 16, and is due by midnight Sep 17. However, the quiz will be open until Sunday midnight to accommodate those students with special extensions. The time limit of the quiz is 65 minutes.

This quiz was locked 19 Sep at 23:55.

Attempt history

	Attempt	Time	Score	Re-graded
LATEST	Attempt 1	65 minutes	13 out of 20	14 out of 20

Score for this quiz: **14** out of 20

Submitted 17 Sep at 21:38

This attempt took 65 minutes.

Question 1

2 / 2 pts

Consider the following code segment:

```
.....
int no_threads, thread_num;
omp_set_num_threads(16);
no_threads=omp_get_num_threads();
#pragma omp parallel
{
    thread_num=omp_get_thread_num();
    if(thread_num==0)
        printf("The number of threads: %d\n", no_threads);
}
```

```
.....
}
```

What is the value of `no_threads` in the `printf()` statement?

Correct!

☒ B. 1

☐ C. 16

☐ A. 0

☐ D. 8

Question 2

2 / 2 pts

What are the data attributes (e.g., *private*, *firstprivate*, *lastprivate*, or *shared*) of variables `result`, `i`, and `nthrds`, respectively, given the following code segment?

```
.....
float result;
int i;
#pragma omp parallel
{
    float B;
    int id, nthrds;
    id = omp_get_thread_num();
    nthrds = omp_get_num_threads();
    for(i = id; i < nthrds; i += nthrds) {
        B = big_job(i);
    }
    #pragma omp critical
        result += calc(B);
}
.....
```

☐ Shared variable: `i`, `result`; private variable: `nthrds`

☐ Private variable: `i, nthrds, result`

☐ Firstprivate variable: `result`; private variable: `i, nthrds`

Correct!

☒ Shared variable: `result`; private variable: `i, nthrds`

Question 3

1.5 / 3 pts

Which of the following for loop parallelizations is (or are) correct?

Correct answer

☐ `#pragma omp parallel for`
`for(int i=k; i<2*k; i++){`
`a[i] = a[i] + a[i-k];`
`}`

☐ `flag = 0;`
`#pragma omp parallel for`
`for(int i = 0; (i < n)&&(!flag); i++) {`
`a[i] = 2.3 * i;`
`if(a[i] < b[i])`
`flag = 1;`
`}`

☐ `#pragma omp parallel for`
`for(int i=1; i<100; i++)`
`a[i] = i*a[i-1];`

```
#pragma omp parallel for reduction(+:dotp)
```

```
{
```

```
    dotp = 0;
```

```
    for(int i = 0; i < n; i++) {
```

```
        dotp += a[i] * b[i];
```

```
    }
```

☐

```
}
```

Correct!

```
#pragma omp parallel for
```

```
for(i=0;i<n;i++){
```

```
    x[i]=a+i*h;
```

```
    y[i]=exp(x[i]);
```

☒

```
}
```

Question 4

Original score: 2 / 3 pts **Re-graded score: 3 / 3 pts**

⚠ This question has been re-graded.

The serial and parallel implementations of a program are given below. Also we know that serial function gives the correct result. Which of the following statements is (or are) correct?

```
// correct serial code
```

```
void serial(float a[], float b[], float c[], int n) {
```

```
    float p,q;
```

```
    int i;
```

```
    for(i=0; i<n; i++) {
```

```
        p = a[i] - b[i];
```

```
        q = a[i] + b[i];
```

```
        c[i] = p*q;
```

```
    }
```

```
}
```

```
// parallel code
```

```
void parallel(float a[], float b[], float c[], int n) {
```

```
    float p,q;
```

```
    int i;
```

```
    #pragma omp parallel for shared(a,b,c,n,p,q) private(i) default(none)
```

```

for(i=0; i<n; i++) {
    p = a[i] - b[i];
    q = a[i] + b[i];
    c[i] = p*q;
}
}

```

Correct!

- ☒ The parallel function could give incorrect result., but not always.

Correct answer

☐

The serial and parallel functions could produce the same results, but not always.

Correct!

☒

Changing the data scope of variables `p` and `q` from `shared` to `private` will give the same result as the serial function.

Correct answer

☐

Changing the data scope of `c` from `shared` to `firstprivate`; as well as the data scope of variables `p` and `q` from `shared` to `private` in the parallel function will give the same result as the serial function.

Question 5

2 / 2 pts

Compared to the serial implementation of the parallel code given below, which of the following is (are) true?

```

#pragma omp parallel for
for(int i=0; i<n; i++) {
    #pragma omp critical
    sum = sum+data[i];
}

```

Correct!

B. The critical directive only allows one thread to execute an iteration of the loop each time so there is no speedup



A. There is a race condition which slows down the program



D. The parallel implementation is faster than the serial implementation



C. The for loop is executed by a number of threads so there is a speedup

Question 6

2 / 2 pts

Given the following program segment,

```
int i, a;
```

```
omp_set_dynamic(0);
```

```
#pragma omp parallel num_threads(4)
```

```
{
```

```
    a = 1;
```

```
    #pragma omp for firstprivate(a) lastprivate(a)
```

```
    for (i=0; i<12; i++)
```

```
        a = a + 2;
```

```
}
```

what is the value of **a** at the exit of the parallel region?



Undetermined



25



5

Correct!

7

Question 7

0 / 3 pts

The following listing gives the code segment of matrix multiplication, $C = A \times B$, where $C \in \mathbb{R}^{N \times N}$, $A \in \mathbb{R}^{N \times N}$, and $B \in \mathbb{R}^{N \times N}$.

```

1 #pragma omp parallel num_threads (NUM_THREADS) shared(A,B,C)
  private(i,j,k)
2   #pragma omp for schedule(static, 4)
3   for(int i=0; i<N; i++)
4     for(int j=0; j<N; j++) {
5       float tmp = 0.0;
6       for(int k=0; k<N; k++)
7         tmp += A[i][k]*B[k][j];
8       C[i][j] = tmp;
9     }

```

We would like to add a `collapse` clause in the parallel code. Which of the following changes is (or are) correct?

Correct answer

☐

Change `#pragma omp for schedule(static,4)` to `#pragma omp for schedule(static,4) collapse(2)`;

You Answered

☒ The proposed changes are all wrong.

☐

Change `#pragma omp for schedule(static,4)` to `#pragma omp for schedule(static,4) collapse(3)`

☐

Change Line `#pragma omp parallel num_threads(NUM_THREADS)` to `#pragma omp parallel num_threads(NUM_THREADS) collapse(2)`

Question 8

1.5 / 3 pts

The following listing gives the code segment of matrix multiplication, $C = A \times B$, where $C \in \mathbb{R}^{N \times N}$, $A \in \mathbb{R}^{N \times N}$, and $B \in \mathbb{R}^{N \times N}$.

```

1 #pragma omp parallel num_threads (NUM_THREADS) shared(A,B,C)
  private(i,j,k)
2   #pragma omp for schedule(static, 4)
3   for(int i=0; i<N; i++)
4     for(int j=0; j<N; j++) {
5       float tmp = 0.0;
6       for(int k=0; k<N; k++)
7         tmp += A[i][k]*B[k][j];
8       C[i][j] = tmp;
9     }

```

Which of the following statements is (or are) correct?

Correct!



If $N = 16$ and $NUM_THREADS = 8$, threads 0, 1, 2, 3 will each get 4 loop iterations from the outermost loop, and threads 4, 5, 6, 7 will get none.

Correct!



Assume $N = 16$ and $NUM_THREADS = 3$. The iteration $i = 14$ in the outermost loop will be executed by thread 0.

You Answered



When $NUM_THREADS=4$, the clause `schedule(static, 4)` in the `for` construct will result in balanced load among threads in the parallel region.



If $N = 16$ and $NUM_THREADS = 8$, each thread will get $N^3/4$ (assuming N^3 divides 4 exactly) loop iterations from the nested loop.

Quiz score: 14 out of 20