**Chose the correct answer**

1. **OpenMP is a ...... level programming model which is ...... programming abstraction.**

a) low, shared memory

b) low, distributed memory

c) high, shared memory

d) high, distributed memory

1. **Multi-thread programs have .... entry point(s) and .... exit point(s).**

a) single, single

b) single, multiple

c) multiple, single

d) multiple, multiple

1. **In Java, a low-priority thread that runs in the background to perform tasks such as garbage collection is called ....**

a) orphan Threads

b) Daemon Threads.

c) Zombie Threads

d) Confused Threads

e) Lonely Threads

1. **Directives are handled in ...... stage.**

a) Prepossessing

b) Compilation

c) Assembling

d) Linking

e) Runtime

1. **Within a parallel region, declared variables are by default** \_\_\_\_\_\_\_\_**.**
2. Private
3. Loco
4. Shared
5. None of the other answers
6. Local
7. **what is the library should be included to use open MP functions**
8. #include<omp.h>
9. #include<paralle.h>
10. #include<openmp.h>
11. #include<mp.h>
12. #include<open.h>
13. **in the end of the parenthesis of**

**#pragma omp parallel**

**{**

**}**

**there is** …………...

1. implicit barrier
2. implicit critical
3. implicit atomic
4. implicit Shared
5. None of the other answers
6. **in the following code:**

**int main()**

**{**

**omp\_set\_num\_threads(3);**

**int id = omp\_get\_num\_thread();**

**}**

**the value of the id is:**

1. 3
2. 2
3. 1
4. 0
5. None of the other answers
6. **OpenMP program is an API for:**
7. shared memory parallel programming
8. distributed memory parallel programming
9. Both of above
10. None of above
11. **in the following code:**

**int main()**

**{**

**omp\_set\_num\_threads(3);**

**int id = omp\_get\_thread\_num();**

**}**

**the value of the id is:**

a) 3

b) 2

c) 1

d) 0

e) None of the other answers

1. **in the following code:**

**int main()**

**{**

**int sum = 10;**

**#pragma omp parallel**

**{**

**sum+=2;**

**}**

**}**

**this code can cause:**

1. false sharing
2. race condition
3. None of the other answers
4. **Directives appear just before a block of code, which is delimited by:**
5. ( … )
6. [ … ]
7. { … }
8. < … >
9. **Which of the following is not considered work sharing construct?**
10. Single
11. Master
12. Section
13. Critical
14. For
15. **The following code will result in a data race:**

**#pragma omp parallel for**

**for (i=1; i < 10; i++)**

**{**

**factorial[i] = i \* factorial[i-1];**

**}**

1. True
2. False
3. **A** \_\_\_\_\_\_\_\_\_\_\_ **construct must be enclosed within a parallel region in order for the directive to execute in parallel.**
4. Parallel sections
5. Critical
6. Single
7. work-sharing
8. **The** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ **specifies that the iterations of the for loop should be executed in parallel by multiple threads.**
9. Sections construct
10. for pragma
11. Single construct
12. Parallel for construct
13. **In OpenMP, assigning iterations to threads is called** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
14. Scheduling
15. Static
16. Dynamic
17. Guided
18. …………….. **initializes each private copy with the corresponding value from the master thread.**
19. firstprivate
20. lastprivate
21. nowait
22. Private (OpenMP) and reduction.
23. **A** …………. **in OpenMP is just some text that modifies a directive.**
24. data environment
25. clause
26. task
27. Master thread
28. **the expected output if we call function omp\_get\_num\_threads() in serial region is .......**
29. runtime error
30. compile error
31. 1
32. 0
33. **Which of the following decides when a task is executed?**

a) runtime system

b) programmer

c) thread

1. **A thread generates a task when it encounters:**

a) task construct

b) parallel construct

c) single construct

1. **What does the nowait clause do?**
   1. Skips to the next OpenMP construct
   2. Prioritizes the following OpenMP construct
   3. Removes the synchronization barrier from the previous construct
   4. Removes the synchronization barrier for the current construct

**True or False**

1. **In shared memory systems, any access from any processing element to the same address has equal latency** (……...….)
2. **In general, Master thread must be the last thread to be terminated, however, in openMP,**

**Master thread can be terminated before their user threads.** (……...….)

1. **PThreads is a distributed memory system.** (……...….)
2. **True or false:** Code in an OpenMP program that is not covered by a pragma is

executed by all threads (……………………..)

1. **T/F : Code in an OpenMP program that is not covered by a pragma is executed by all threads.** (……….)
2. **there is implicit barrier at the end of master construct**. (……...….)
3. **Name at least one difference between master construct and single construct?**
4. **The master region can be executed by any thread including the master thread.** (……...….)
5. **T/F: if we declared any variable in the sequential part of the program**

**then it can only be shared among all threads.** (………….)

1. **T/F: If the data-sharing attribute of a variable is private within a construct, a separate local copy of the same variable is created for every thread (including the master thread).** (……….)
2. **the thread can change its own ID (THREAD\_NUM) during execution.** (……...….)
3. **can multiple threads have same ID (THREAD\_NUM) in Nested parallelism.** (……...….)
4. **we must initialize the Enviroment variables At the beginning of the program to use it .** (…………..)
5. **we could override the default value of the Environment variable (OMP\_NUM\_THREADS)**

**inside the program.** (…………..)

1. **The default value of the environment variable (OMP\_NUM\_THREADS) is the number of processors in your machine.** (…………..)
2. **we couldn't enter the single region with more than one thread.** (…………..)

**Code Questions**

**int i;**

**double sum;**

**// sum = 1;**

**#pragma omp parallel for reduction(\* : sum)**

**for (i=1; i <= 4; i++)**

**sum = sum \* i;**

**printf(“The sum is %lf \n”,sum);**

**Do we have to uncomment line 3 so that my code run correctly? give a reason.**

**int i;**

**double sum = 0.0;**

**// sum = 1;**

**#pragma omp parallel for firstprivate(sum) num\_threads(1)**

**for (i=1; i <= 4; i++)**

**sum = sum + 1;**

**printf(“The sum is %lf \n”,sum);**

**what is the value of (sum) after executing this code?**

1. **<if we disable the nested parallelism ,and using the following construct>**

**#paragma omp parallel num\_threads(3)**

**{**

**//region 1**

**#paragma omp parallel num\_threads(4)**

**{**

**//region 2**

**}**

**}**

1. **the number of threads working in region 1 (at the same time ) is:**
2. 1
3. 2
4. 3
5. 4
6. **the number of threads working in region 2 (at the same time ) is:**
7. 1
8. 2
9. 3
10. 4
11. **number of teams working in region 1 (at the same time ) is**
12. 1
13. 2
14. 3
15. 4
16. **number of teams working in region 2 (at the same time ) is:**
17. 1
18. 2
19. 3
20. 4

**4) In the flowing 2 versions of a program to execute 2 tasks:**

#pragma omp parallel

{

#pragma omp single nowait

{

#pragma omp task

b = beta();

#pragma omp task

a = alpha();

}

}

#pragma omp parallel

{

#pragma omp single nowait

{

#pragma omp task

b = beta();

a = alpha();

}

}

1. Why in the second pragma, nowait is used ?
2. What is the difference between the 2 versions ?
3. **Variables: A=1 ; B=1 ; C=1**

**#pragma omp parallel private(B) firstprivate(C)**

Are A,B,C local to each thread or shared inside the parallel region?

What are their initial values inside?