CS528-Quiz 1

CS528-Quiz 1

Points: 25/36

1. Roll No *

190101056

2. Name *

Mushanolla Pranathi

✓ **Correct** 1/1 Points

3. Which level of GCC compiler optimization support vectorization or simidization

- Level 0
- Level 1
- Level 2
- Level 3 ✓
- Non of the above

✓ **Correct** 2/2 Points

4. Tick the correct statement about loop parallelization using simidization process of the given loop.

```
for (i=0; i<N; i++){
  p=3*i+4; q=6*i+7; r=2*i*i+25;
  X[p]=X[q]+r;
}
```

- ( ) Parallelization is definitely possible and there is no dependency between iteration
- ( ) Parallelization is may be possible and there may be not having dependency between iterations
- (•) Parallelization is not possible and there is a dependency between iteration ✓
- ( ) We cannot say

✓ **Correct** 1/1 Points

5. Every thread of a process

- (•) Shares code, data space, and kernel context with other threads but have their own logical control flow and stack areas for variables and function calls. ✗
- ( ) Shares code and data space with other threads but have their own logical control flow, kernel context, and stack areas for variables and function calls.
- ( ) It Shares data space and kernel context with other threads but have their own code logical control flow and stack areas for variables and function calls. ✓
- ( ) None of the above.

✓ **Correct** 1/1 Points

6. Good thread-safe functions and thread-safe data structure are used to implement multithreaded applications: All these thread safe functions and thread safe DS ensure

- ○ High performance
- ◉ Ensure data safety ✗
- ○ Endure both performance and data safely ✓
- ○ Not of these.

✓ **Correct** 1/1 Points

7. Row major access of matrix is better as compared to column-major access of the matrix as it exhibits ____.

- ◉ Spatial locality ✓
- ○ Temporal locality
- ○ Both spatial and temporal locality
- ○ We cannot say.

✓ **Correct** 1/1 Points

8. Avoiding branches in loop code results

- ○ Reduce scope of simidization but enable software pipelining
- ○ Increase the scope of simidization but disable software pipelining
- ◉ Enable the simidization to use the AVX and can also be pipelined using software pipelining ✓
- ○ None of the above.

✓ **Correct** 2/2 Points

9. Calculate code balance of the following code in B/F, (assume write allocate)

```
for(i=0;i<N;i++) //double a[], b[], s
    a[i]=a[i]+s*b[i];
```

12

✗ **Incorrect** 0/2 Points

10. Given a cache of size 1MB with 4 way set associative and block (or line) size of 64B, calculate the number of bits required for the index field

9

Correct answers: 12

✓ **Correct** 1/1 Points

11. What are typical the benefits of anti-aliasing the array access during function declaration?

- Can be vectorized the loop
- Can apply software pipelining on the loop
- Can perform group load-store transformation in the loop
- All of the above ✓

✓ **Correct** 1/1 Points

12. Data coherence problem in a multiprocessor system is about

- ○ All the cores need to execute the same program
- ○ All the cores need to run at the same frequency
- ◉ All the places value of the considered shared variable need to be the same. ✓
- ○ All the cores have to finish the program at the same time.

✗ **Incorrect** 0/2 Points

13. Which of the following type of applications cannot be categorized as compute-intensive, cache-friendly, computation up-loadable (acceleration using GPU/Cloud) friendly application

- ☐ O(N^2)/O(N^2) class (Ops/Data) ✓
- ☑ O(N)/O(N^2) class (Ops/Data) ✓
- ☐ O(N$3)/O(N) class (Ops/Data)
- ☐ O(N^4)/O(N) class (Ops/Data)

✓ **Correct** 1/1 Points

14. Choose the right answer

- OpenMP, Pthread can run on one computer and multicomputer whereas MPI can run only multi-computer
- OpenMP, Pthread can run on only one computer whereas MPI can run both one computer and multi-computer ✓
- OpenMP, Pthread can run on multicomputer whereas MPI can run both multi-computer and a single computer
- OpenMP, Pthread, and MPI can run on run both one computer and multi-computer

✓ **Correct** 1/1 Points

15. Which of the tool among these can be used to generate a graphical call graph profile for an application

- Valgrind ✓
- Gprof
- Gcov
- GCC

✓ **Correct** 2/2 Points

16. Is this loop is parallelizable using Pthread?

```
void VectorAvg(){
  for(int j=1;j<SIZE-1;j++)
  A[j]=(B[j-1]+B[j]+B[j+1])/3.0;
 }
```

- ( ) Maybe
- ( ) May not be
- (•) Yes ✓
- ( ) NO

✗ **Incorrect** 0/1 Points

17. Given OMP code with 4 threads. Suppose the run time function ReL is proportional to 1/i. Which of the scheduling strategy will be beneficial for better performance?

```
#pragma omp parallel for schedule (X, Y)
 {
   for (i=1; i<1000; i++)  A[i]=ReL(i);
 }
```

- (•) X = static, Y=10 ✗
- ( ) X = static, Y=20
- ( ) X = dynamic, Y=10
- ( ) X = guided, Y=10 ✓

✗ **Incorrect** 0/1 Points

18. What is the best possible optimization, we can think of the following code [hint: code don't take any external input]

```
int X=0, Y=0; N=20;
 For (i=1;i<=N;i++) X=X+i;
 For (i=1;i<=N;i++) Y=X+i*i;
 printf("X=%d, Y=%d",X,Y);
```

- ( ) Merge both the for loop
- ( ) Use AVX and Simidization
- ( ) Use copy propagation and static calculation ✓
- (•) All of the above ✗

✗ **Incorrect** 0/2 Points

19. Energy consumption of system running at one-third of maximum capable frequency will be around ___ times lesser energy consumption of the same running at maximum capable frequency?

1/27

Correct answers: 9

✓ **Correct** 2/2 Points

20. Given Ppeak and bs a machine 4 GF/s and 10 GB/s respectively, what will be the achieved performance of the following loop in Giga flop per second. (assume write allocate)

```
for(i=0;i<N;i++) s=s+a[i]*a[i]; //float s, a[N]
```

4 GF/s

Correct answers: 4

✓ **Correct** 1/1 Points

21. Why multiprocessor is popular as compared to a single processor

- ○ Cost and performance effective
- ○ Power effective
- ○ Energy effective
- ◉ All the above ✓

✓ **Correct** 2/2 Points

22. Given the following kernel, the best way to speed up this in Modern day processors of a single machine (PC/Laptop)

```
unsigned char x[N], w[N], k;
for(i=0;i<N;i++)   x[i]=(w[i]*x[i])-k;
```

- ○ Using AVX
- ◉ Using OpenMP and AVX ✓
- ○ Using MPI
- ○ None of the above

✓ **Correct** 1/1 Points

23. Which of the following tools can be used for Hot Spot Analysis?

- Gprof
- Gcov
- Valgrind
- All of the above ✓

✓ **Correct** 2/2 Points

24. Choose the correct statement for optimizing by the compiler?

- Optimization is limited to within a function, use only static information, and when doubt compiler needs to be conservative ✓
- Optimization in general use whole program code, with the dynamic view and very aggressive
- Optimization, in general, uses whole program code, with the static view, and when doubt compiler needs to be conservative
- All the above

✗ **Incorrect** 0/1 Points

25. Why heterogeneous multiprocessor SoC is better as compared to homogeneous multiprocessor SoC?

- (●) Homogeneous multiprocessor SoC have all generic cores may not be best for specific tasks ✗
- ( ) Homogeneous cores of SoC may not be energy efficient
- ( ) As SoC is used for much specialized works, generalized cores in homogeneous systems do not give better power and performance tradeoff
- ( ) All of the Above ✓

✓ **Correct** 2/2 Points

26. Given a direct mapped cache with 64B line size and 1024 sets, calculate total number of number miss to be occurred for the array accesses of the following loop. Assume address A started from 00000 and scalar i is mapped to register.

```
unsigned char int A[1024];
 for(i=0;i<1024;i++) A[i]=i;
```

64

Correct answers: 16

✕ **Incorrect** 0/2 Points

27. Suppose you want to optimize the code using LUT; How many times you can avoid calling the transcendental functions (sin/cos).

```
double S,X;
  X=getInputFromKeyBoard();
  for(i=0;i<10000;i++) S=S+cos(X/(i%10))+ sin(X/(10-i%10));
```

1000

Correct answers: 19980