1. Which keyword initializes each private copy of a variable with the corresponding value from the master thread.

[redacted]
Firstprivate
Private
Shared
Default

2. What does the **nowait** clause do?

[redacted]
Skips to the next OpenMP construct
Prioritizes the following OpenMP construct
Removes the synchronization barrier from the previous construct
[✓ Removes the synchronization barrier for the current construct]

3. The default clause sets the default scheduling of threads in a loop construct.

[redacted]
True
[✓ False]

4. Assume that you have 10 cores that you can use to solve a problem in parallel - 98% of your code is parallelizable. Can you get a speedup of 7? If so, how many cores(minimum) are needed?

speed 98

a) 6 Cores
b) 7 Cores
c) 8 Cores
d) 9 Cores

5. Answer which is most correct about non-blocking sends, i.e., MPI_Isend(...)

[redacted]
As soon as the send returns the data being sent can be modified by the sender process.
MPI_test can be used to block the process until the send has completed.
MPI_wait can be used to block the process until the send has completed.
all MPI communication statements that send data ensure that the data being sent is safe.

6. Assume we have a problem where we want to find the sum of an array of size 10000.You have 2 solutions for the given problem. First solution is the serial version of the code and other solution is the parallel version of the code using 1 thread. Which will run faster?

[redacted]
Serial Solution
Parallel Version
Both are same

7. What will be the output of following code

```
int main()
{
 int i;
 const int N= 5;
 int a= 50;
 int b= 0;
omp_set_num_threads(5);
#pragma   omp   parallel   for   default(none)   private(i)   firstprivat
```

```
lastprivate(b)
 for(i=0; i<N; i++) {
      b = a + i; }
 printf("a=%d b=%d \n", a, b);
```

[redacted]
a=50 b=54
a=50 b=4
a=0 b=54
The code will generate an error

8. Which of the following MPI function is non-blocking

[redacted]
MPI_WAIT
MPI_TEST
MPI_Probe
MPI_Recv

9. We have an array of size 15 with values [13,4,5,1,2,3,44,15,16,7,8,9,78,65,36] What will be the values assigned to each to the process having rank 2 if displs array [0,5,9,12] and sendcounts array [5,3,2,4] is passed to MPI_ScatterV function.

```
int MPI_Scatterv(const void *sendbuf, const int *sendcounts, const int
*displs,MPI_Datatype sendtype, void *recvbuf, int recvcount,
MPI_Datatype recvtype,int root, MPI_Comm comm)
```

[✓ a)] [7,8]
b) [13,4,5,1,2,3,44,15,16]
c) [16,7,8]
d) [0,5]

10. what will be the output of the following OpenMP code? Assume that there are 4 threads all together:

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[]) {
 int i;
double sum = 0.0;
#pragma omp parallel private (sum)
{
for (i=1; i <= 4; i++)
sum = sum + 1;
}
printf("The sum is %1f\n", sum);
}
```

[redacted]
The sum is 0.0
The sum is 4.0
The sum is 16.0
The code will generate an error

```
HelloWorld.cpp: In function 'int main(int
HelloWorld.cpp:6:1: error: 'MPI_Status' w
```

014

National University of Computer and Emerging Sciences

FAST School of Computing Spring-2022 Islamabad Campus

Question 2 [6 Marks]

An MPI program has data (input buffer) that is shown in the figure below. For Parts (a to c) provide the name of the collective communication operation that will lead to the corresponding output

P_0	1	2	3	4	5
P_1	6	7	8	9	10
P_2	11	12	13	14	15
P_3	16	17	18	19	20
P_4	21	22	23	24	25

part	Output Buffer					Collective Communication Operation (Only Mention the name)
a)	P_0	55	60	65	70	75	[redacted]
	P_1	55	60	65	70	75	
	P_2	55	60	65	70	75	
	P_3	55	60	65	70	75	
	P_4	55	60	65	70	75	

b)

P_0	1	6	11	16	21
P_1	2	7	12	17	22
P_2	3	8	13	18	23
P_3	4	9	14	19	24
P_4	5	10	15	20	25

MPI_Scatterv

✗

c)

P_0	1	2	3	4	5
P_1	1	2	3	4	5
P_2	1	2	3	4	5
P_3	1	2	3	4	5
P_4	1	2	3	4	5

MPI_ Bcast

2

0148

National University of Computer and Emerging Sciences

FAST School of Computing   Spring-2022   Islamabad Campus

Question 3 [12+8 Marks]

a) For each loop in the following code, state whether or not it is parallel, and if not describe the dependence(s) that prevent it from being parallel (give the type of dependence and which terms c1,c2,c3 are involved).

```
for (i=2;i<N-1;i++){ // loop1
        for (j=3;j<N;j++){ // loop2
            for (k=4;k<N-3;k++){ // loop3
                A[i][j][k] =  // c1
                A[i+1][j][k+2] + // c2
                A[i][j][k-1]; // c3
            }
        }
}
```

Answer

4

Loop can be Parallel (Yes/No)	NO

Dependency *(Mention the statements having dependency)*	**Type of Dependency**
k=4;k<N-3;k++ A[i][j][k] → C1	True dependency.
A[i][j][k]=... C2	True dependence
C3	anti dependency

0

2

names not correct

b) Suppose a program has a part at the beginning that is sequential in nature (must be executed by only one processor) and takes 3ms. Also, there is a part at the end of the program that is sequential (must be executed by only one processor) and takes 4 ms. Rest of the code is divided into 5 equal parts that are executed in parallel on 5 processes and each of these parts takes 16 ms. Calculate speedup, scaled speedup and efficiency.

[illegible]

[illegible]

Speed Up	12
Efficiency	43.75

Scaled Speedup	0.136 7
Efficiency	8.5

ough Work:

Serial part → 3ms

Serial part → 4ms

parallel = 5 = 16ms ~~5~~/process 1

time of 1 process / time multiprocess

$$\frac{3+4}{16*5}$$

$$\text{Scaled} = \frac{r\ ts+tp}{1-tp} \qquad \frac{1}{1-tp}$$

$$r = \frac{1}{x} = 0.0625$$

Question 4 [8 Marks]

Complete the following sketch program to compute the sum of all numbers given in an array using OPENMP. If you have multiple solutions in mind write the one having best performance.

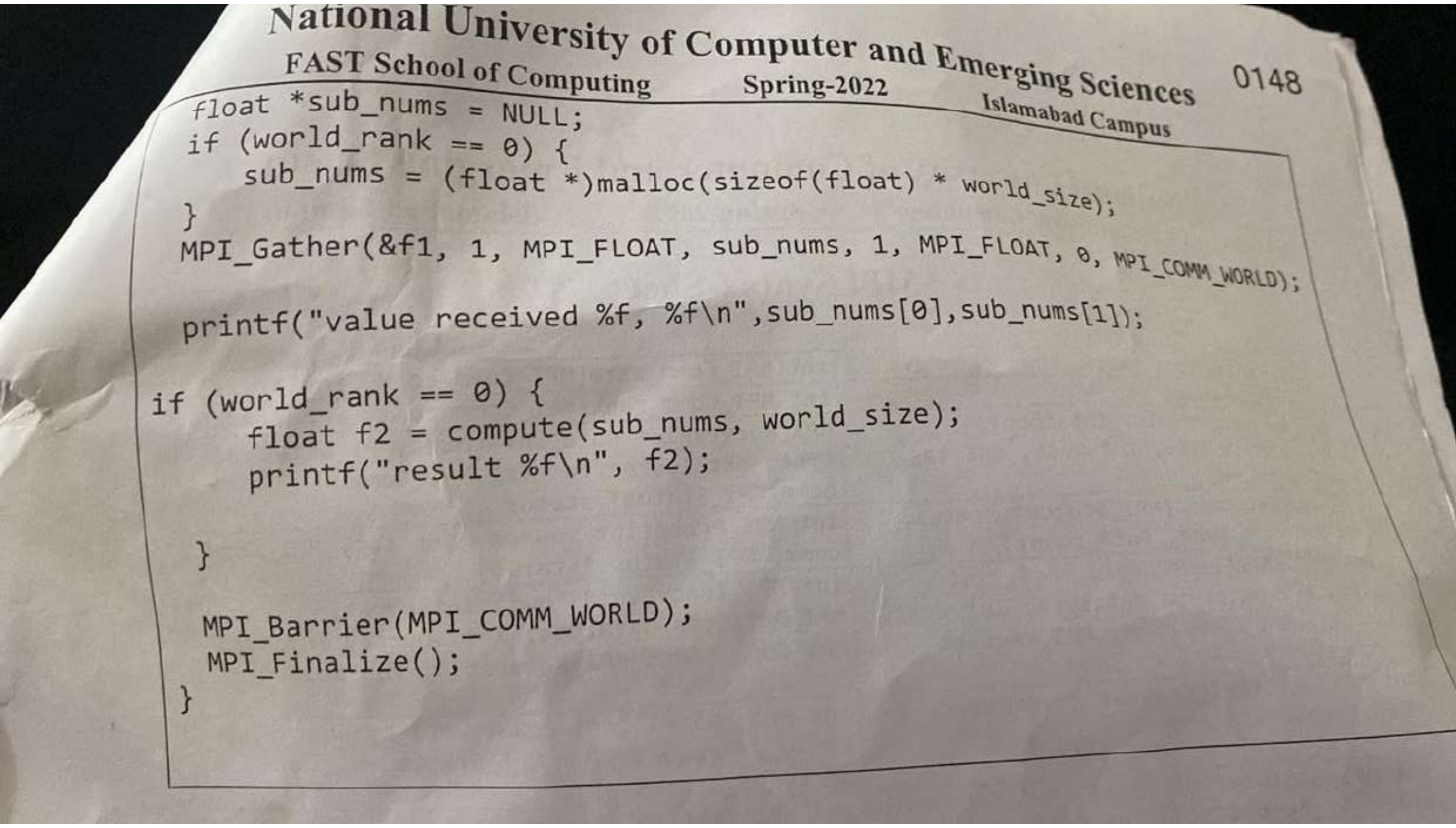
```
int main() {

  const int N=100;
  int a[N];

  //initialize
  for (int i=0; i < N; i++)
    a[i] = i;

  //compute sum
  int local_sum, sum;        int th
//Add your code here
~~#pragma omp parallel~~ = omp_set_num threads(10)
              #pragma omp parallel (th)
  {                                         5
    for (int i=0 ;i<N; i++)
      local_sum = local_sum+a[i]

    }

    sum = local_sum ;




intf("sum=%d should be %d\n", sum, N*(N-1)/2);
```

Question 5 [6 Marks]

Show the output of the following MPI code.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <mpi.h>
#include <assert.h>
                                        5
float compute(float *array, int num_elements) {
  float sum = 0.f;
  int i;
  for (i = 0; i < num_elements; i++) {
     sum += array[i];
  }
 return sum / num_elements;
}

int main() {

  int num_elements_per_proc=5;
  MPI_Init(NULL, NULL);

  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);   // getting rank
  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);   // getting size.
                                                 8
  float *sub_rand_nums = (float *)malloc(sizeof(float) *
num_elements_per_proc);
 float rand_nums[10]; ——————————————————→ elements.
 if (world_rank == 0) {                          2 threads
    for (int i=0;i<10;i++){                      5 per thread
         rand_nums[i] =i+10;
         printf("%f\n",rand_nums[i]);
    }
         thread 1:   10, 20, 30, 40, 50
}


   MPI_Scatter(rand_nums, num_elements_per_proc, MPI_FLOAT,
sub_rand_nums, num_elements_per_proc, MPI_FLOAT, 0, MPI_COMM_WORLD);

 // Compute the average of your subset        5
                      4x5=20
 float f1 = compute(sub_rand_nums, num_elements_per_proc);
 printf("%f\n",f1);
```

Byte word



```
float *sub_nums = NULL;
if (world_rank == 0) {
    sub_nums = (float *)malloc(sizeof(float) * world_size);
}
MPI_Gather(&f1, 1, MPI_FLOAT, sub_nums, 1, MPI_FLOAT, 0, MPI_COMM_WORLD);

printf("value received %f, %f\n",sub_nums[0],sub_nums[1]);

if (world_rank == 0) {
     float f2 = compute(sub_nums, world_size);
     printf("result %f\n", f2);


  }

  MPI_Barrier(MPI_COMM_WORLD);
  MPI_Finalize();
}
```