National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	**Parallel and Distributing Computing**	**Course Code:**	**CS3006**
	Degree Program:	**BS (CS)**	**Semester:**	**Spring 2023**
	Exam Duration:	**60 Minutes**	**Total Marks:**	**35**
	Paper Date:	**--/04/23**	**Weight**	**12.5**
	Exam Type:	**Mid II Retake**	**Page(s):**	**5**

Student : Name:_________________________ Roll No.________________ Section:_______

Question # 1 a: MCQs [6 marks, CLO # 1]

From the given options, select the best answer.

i. **When calculating message passing costs, we would assume that "tw" would be effected by?**
   a. Switch latencies
   b. Number of hops
   c. Time needed to add headers
   d. Bandwidth of the links

ii. **For a 2*16 mesh, using a naïve solution, how many messages would you expect the sending process to send, for a 1-to-all broadcast?**
   a. Less than 22
   b. More than 30
   c. Less than 10
   d. Less than 16

iii. **With all-to-1 reduction, we are essentially?**
   a. Requiring all (p-1) processes to send messages to all p process
   b. Requiring 1 process to send messages to (p-1) processes
   c. Requiring all (p-1) processes to send messages to 1 process
   d. Requiring all p processes to send messages to 1 process

iv. **Which of the following is not an issue in distributed systems?**
   a. Programming complexity
   b. Scalability
   c. Scarce robustness
   d. Hard to optimize

v. **Simple Storage Service (S3) is an example of:**
   a. Infrastructure as a Service
   b. Platform as a Service
   c. Software as a Service
   d. None of the given options

Question # 1 b: [3 marks, CLO # 1]

True/False

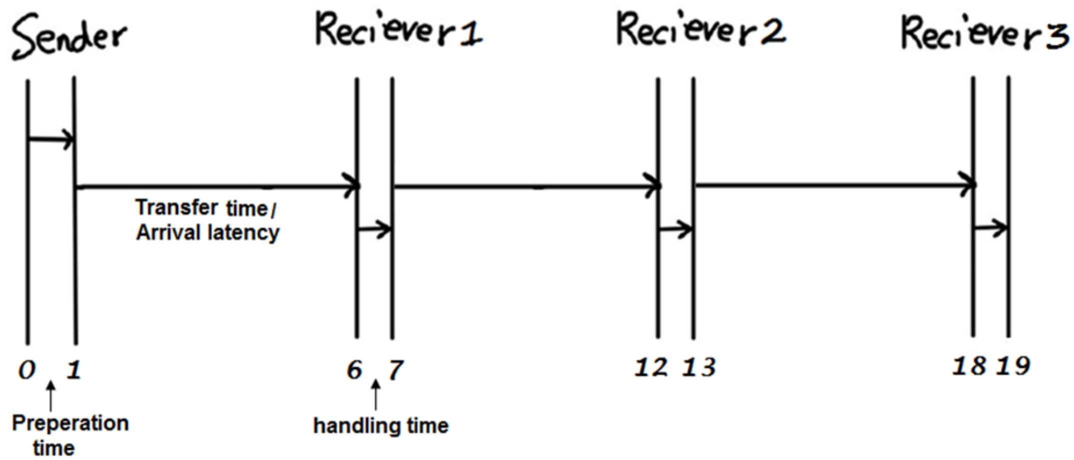
I. **Distributed Operating Systems are better suited for heterogeneous multi-computers.**
   a. True
   b. False

II. **Cluster Computing is a form of Utility Computing.**
   a. True
   b. False

III. **Grid Computing is biased towards general purpose computing.**
   c. True
   d. False

Question # 1 c: [6 marks, CLO # 1]

Calculate the time required to transfer 400 mbits of data from Sender to Receiver3. Bandwidth of the link is 10 mbits/s.



$$t = t_s + (t_w \cdot m + t_h + t_r)\, n$$

t_s = Preparation Time = $t_1 - t_0 = 1 - 0 = 1$ s

t_w = 1 unit transfer time = $\frac{1}{r} = \frac{1}{10} = 0.1$

m = message size = 400

t_h = Arrival Latency = $t_2 - t_1 = 6 - 1 = 5$

t_r = Receiver Handling Time = $t_3 - t_2 = 7 - 6 =$ [illegible]

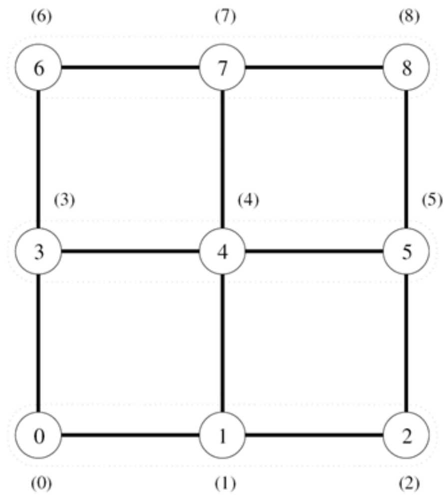
n = # of hops = 3

$$t = 1 + (0.1(400) + 5 + 1)\, 3$$

$$\boxed{t = 139\,s}$$

Question # 2: [4+3+3 marks, CLO # 3]

a) **Explain the All-reduce communication operation on the following mesh with 9 nodes.**



b) Provide total cost estimation for this operation. You have to consider the size of each message!

one to all bcast = log(P)(ts+mtw)
all to one red = same

total = 2*log(P)(ts+mtw)

c) What modification would be required in the following code to perform All-reduce operation?

procedure ALL_TO_ALL_BC_MESH(my_id, my_msg, p, $result$)
begin

/* Communication along rows */
$left := my\_id - (my\_id \bmod \sqrt{p}) + (my\_id - 1) \bmod \sqrt{p}$;
$right := my\_id - (my\_id \bmod \sqrt{p}) + (my\_id + 1) \bmod \sqrt{p}$;
$result := my\_msg$;
$msg := result$;
for $i := 1$ **to** $\sqrt{p} - 1$ **do**
send msg to $right$;
receive msg from $left$;
$result := result \cup msg$; result = result+msg
endfor;

/* Communication along columns */
$up := (my\_id - \sqrt{p}) \bmod p$;
$down := (my\_id + \sqrt{p}) \bmod p$;
$msg := result$;
for $i := 1$ **to** $\sqrt{p} - 1$ **do**
send msg to $down$;
receive msg from up;
$result := result \cup msg$;
endfor;
end ALL_TO_ALL_BC_MESH

Question # 3: [6+4 marks, CLO # 2]

a) **Write the output for the following piece of code assuming that there are 4 MPI processes. Assume there is no syntax error.**

```
#include <mpi.h>
#include <stdio.h>

int main (int argc, char** argv) {
MPI_Init (NULL, NULL);
MPI_Status status;

int p, b, my_rank;
MPI_Comm_size(MPI_COMM_WORLD, &p); //NO of MPI process     p=4
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);  //pids, label

int a = my_rank+10;
int sTag = my_rank;
int rTag = (my_rank - 2 + p) % p;
int next = (my_rank + 2) % p;
int prev = ((my_rank - 2 + p) % p);

MPI_Sendrecv(&a,1,MPI_INT,next,sTag, &b,1,MPI_INT,prev,rTag, MPI_COMM_WORLD, &status);
printf("I am %d: Got:%d from %d and Sent:%d to %d\n ", my_rank, b, prev, a, next);
MPI_Finalize();
}
```

Status{
MPI_SOURCE,
MPI-TAG,
MPI_ERROR}

MPI_ANY_SOURCE

WILDCARD ENTRY

rank= 0;
I am 0 Got:12 from 2 and Sent 10 to 2
I am 1 Got: 13 from 3 and Sent 11 to 3
I am 2 Got: 10 from 0 and Sent 12 to 0
I am 3 Got: 11 from 1 and Sent 13 to 1

b) **Describe one reasonable scenario where we would use MPI_ANY_SOURCE?**

WILD CARD ENTRY

c) **When we want all processes to synchronize, which MPI function should we call?**

MPI_BARRIER