Question 1 [Multiple choice questions]
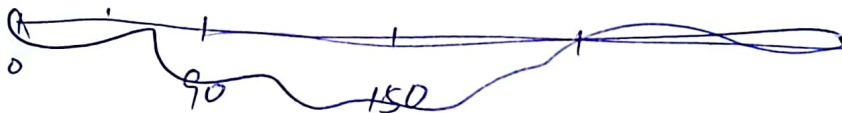
Please use the following process table for the first 15 MCQs. Q1-Q5 contain questions about Shortest Job First, Q6-Q10 about Shortest Remaining Time First and Q11-Q15 about Round Robin with q=10ms.

Process	Arrival Time (ms)	CPU Time (ms)
A	0	90
B	10	60
C	30	150
D	50	40
E	70	70

Rough work space for SJF is given below. Rough Work is mandatory and must match your answers. Besides this given space, you may use space in front of MCQs for extra rough work (if needed)

1. Using SJF, what is the waiting time for Process B?
   a. 100
   b. 110
   c. 120
   d. 130
   e. None of the above

2. Using SJF, what is the waiting time for Process E?
   a. 100
   b. 110
   c. 120
   d. 130
   e. None of the above

3. Using SJF, what is the turnaround time for Process E?
   a. 170
   b. 180
   c. 190
   d. 200
   e. None of the above

4. Using SJF, what is the average waiting time for the example process set?
   a. 98
   b. 100
   c. 102
   d. 104
   e. None of the above

5. Using SJF, what is the average turnaround time for the example process set?
   a. 180
   b. 184
   c. 190
   d. 194
   e. None of the above

Rough work space for SRTF is given below. Rough Work is mandatory and must match your answers.
Besides this given space, you may use space in front of MCQs for extra rough work (if needed)

	A.T	T.A	W.T	Avg W.T = 92
A	0	260	170	
B	10	60	0	
C	30	380	230	Avg = 7A = 876/5 = 174
D	50	60	20	
E	70	110	40	

AR :- 80

80

Q

A   B

80
↑
A → 10   40
↑
B → 30 + 20
C → 150

10₂   20   30   40   50   60   70   80   90   100   110   180 260

CAYA   DY   B·D   B·E   E·A A·C

6. Using SRTF, what is the turnaround time for Process B?
   - a. 50
   - b. 60
   - c. 70
   - d. 80
   - e. None of the above

7. Using SRTF, what is the turnaround time for Process D?
   - a. 50
   - b. 60
   - c. 70
   - d. 80
   - e. None of the above

8. Using SRTF, what is the turnaround time for Process E?
   - a. 100
   - b. 120
   - c. 140
   - d. 160
   - e. None of the above

9. Using SRTF, what is the average waiting time for the example process set?
   - a. 90
   - b. 94
   - c. 98
   - d. 102
   - e. None of the above

10. Using SRTF, what is the average turnaround time for the example process set?
    - a. 170
    - b. 174
    - c. 178
    - d. 182
    - e. None of the above

Rough work space for RR (with q= 10 ms) is given below. Rough Work is mandatory and must match your answers. Besides this given space, you may use space in front of MCQs for extra rough work (if needed)
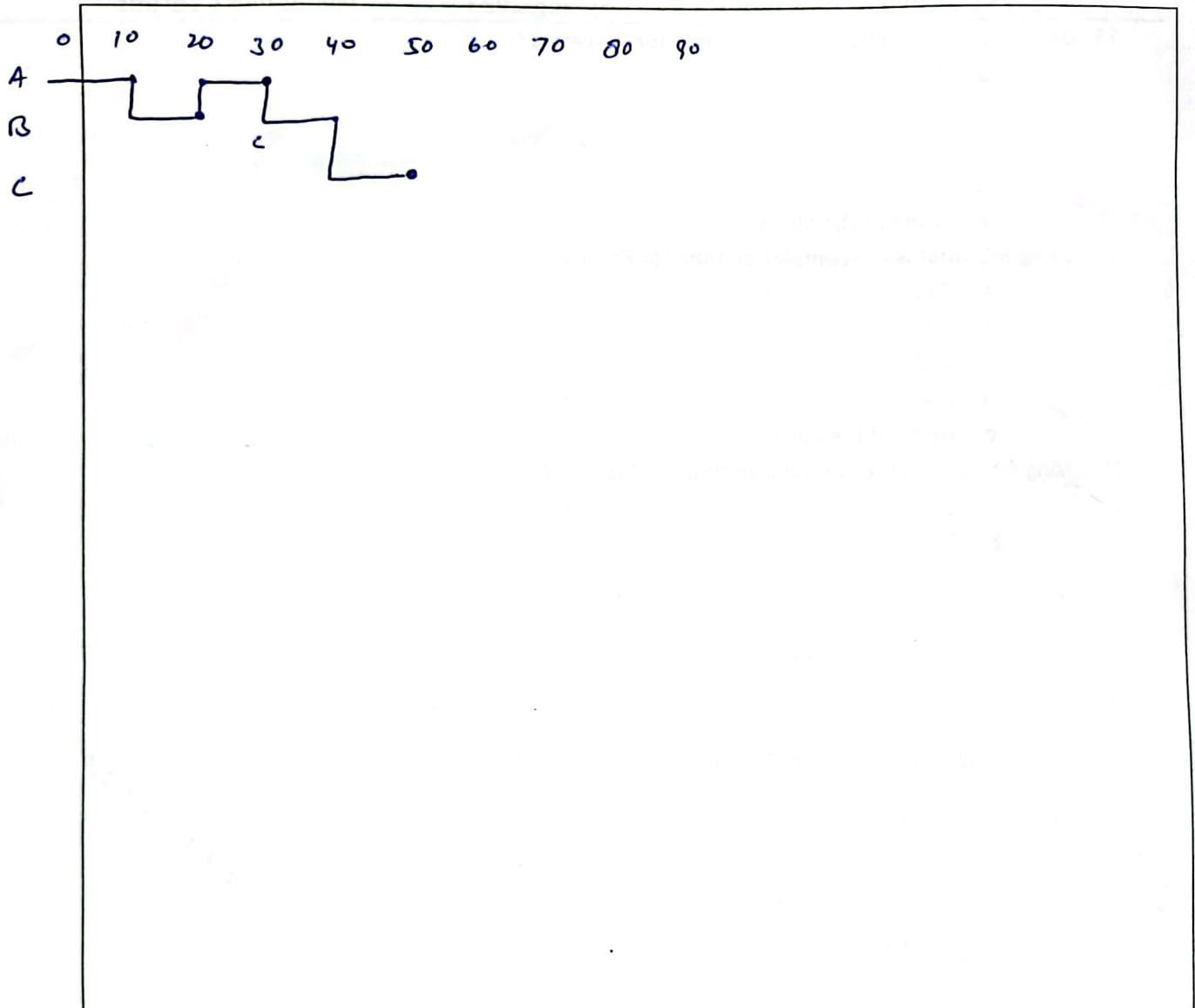
11. Using RR, what is the completion time for Process A?
   a. 270
   b. 280
   c. 290
   d. 300
   e. None of the above

12. Using RR, what is the completion time for Process B?
   a. 190
   b. 200
   c. 210
   d. 220
   e. None of the above

13. Using RR, what is the completion time for Process C?
    a. 380
    b. 390
    c. 400
    d. 410
    e. None of the above

14. Using RR, what is the completion time for Process D?
    a. 210
    b. 220
    c. 230
    d. 240
    e. None of the above

15. Using RR, what is the completion time for Process E?
    a. 350
    b. 360
    c. 370
    d. 380
    e. None of the above

For the next 10 MCQs, use the following process table.

Process	Arrival (at ms)	Threads	Service Time (in ms)
P1	0	T1	5
	0	T2	4
P2	2	T1	3
	2	T2	3
	2	T3	4
	2	T4	5
P3	3	T1	3
	3	T2	4
	3	T3	3
	3	T4	6
P4	5	T1	7
	5	T2	3
	5	T3	5

For the above given table, consider the following information about a set of independent processes and their threads. Related/cooperating threads per process are shown, each thread has its different execution/service time. Assume a multi-processor system uses gang scheduling on **FOUR CPUs** (C1,C2,C3 & C4). The scheduling scheme is Uniform Weighted Scheduling and the time slice for each gang is **2 ms**.

CS CamScanner

Rough work space for SJF is given below. Rough Work is mandatory and must match your answers. Besides this given space, you may use space in front of MCQs for extra rough work (if needed)
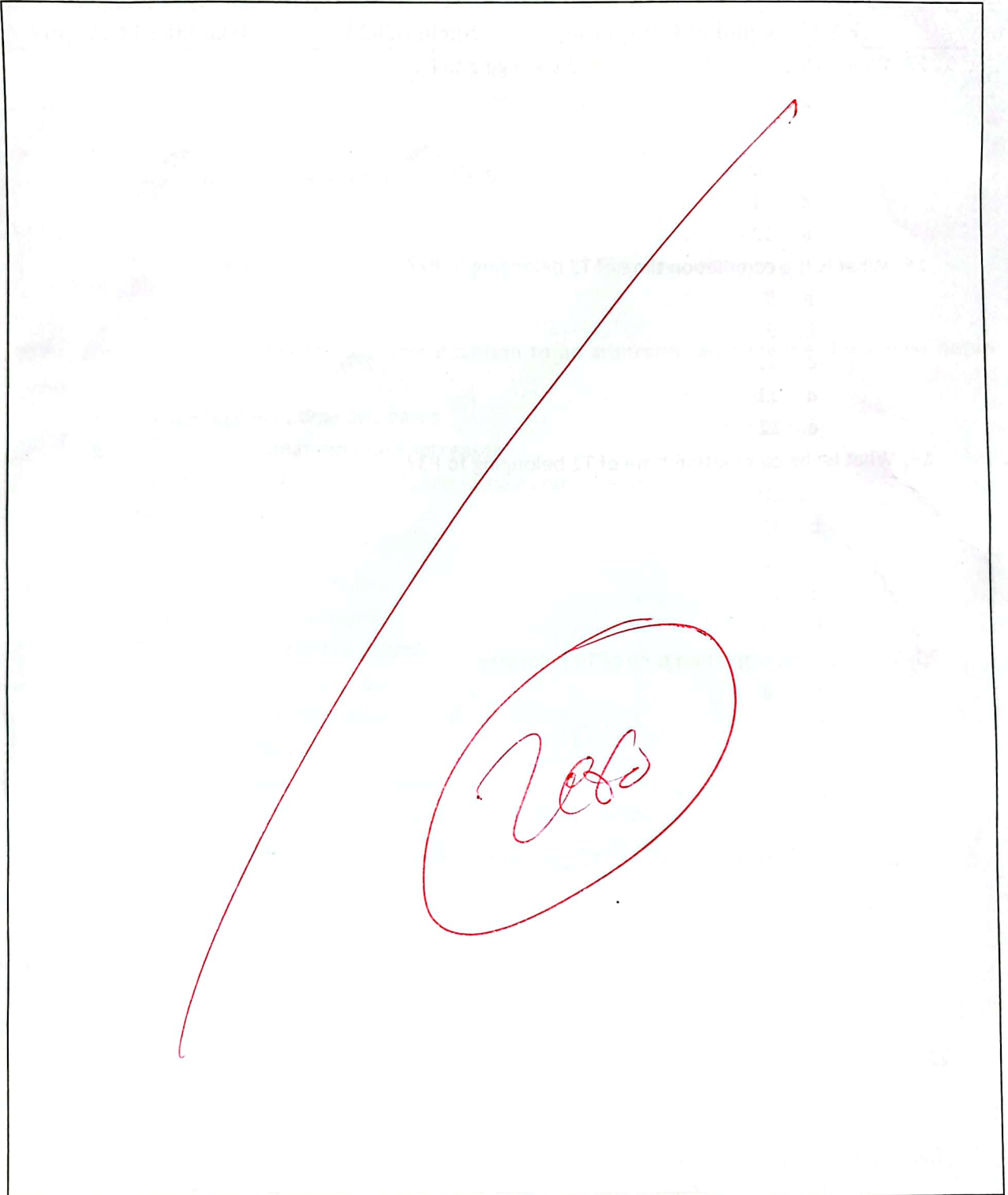
16. What is the completion time of T1 belonging to P1?
    a. 3
    b. 4
    c. 5
    d. 6
    e. 7

17. What is the completion time of T1 belonging to P2?
    a. 8
    b. 9
    c. 10
    d. 11
    e. 12

18. What is the completion time of T3 belonging to P2?
    a. 8
    b. 9
    c. 10
    d. 11
    e. 12

19. What is the completion time of T2 belonging to P3?
    a. 10
    b. 11
    c. 12
    d. 13
    e. 14

20. What is the completion time of T4 belonging to P3?
    a. 18
    b. 19
    c. 20
    d. 21
    e. 22

21. What is the completion time of T2 belonging to P4?
    a. 16
    b. 17
    c. 18
    d. 19
    e. 20

22. What is the completion time of T3 belonging to P4?
    a. 16
    b. 17
    c. 18
    d. 19
    e. 20

23. When does the Process P2 ends
    a. 13
    b. 14
    c. 15
    d. 16
    e. 17

24. When does the Process P3 ends
    a. 16
    b. 17
    c. 18
    d. 19
    e. 20

25. When does the Process P4 ends
    a. 20
    b. 21
    c. 22
    d. 23
    e. 24

26. What is the effect of increasing the time quantum to an arbitrarily large number for Round Robin scheduling?
    a. The average wait time decreases
    b. The average turnaround time increases
    c. The algorithm behaves exactly the same as non-preemptive FCFS
    d. All of the above
    e. None of the above

27. Termination of the process terminates
    a. First thread of the process
    b. First two threads of the process
    c. All threads within the process
    d. All threads except the main thread
    e. No thread within the process

28. Thread synchronization is required because:
    a. all threads of a process share the same address space
    b. all threads of a process share the same global variables
    c. all threads of a process can share the same files
    d. both a & b
    e. a, b, & c

29. Multithreading an interactive program will increase responsiveness to the user by :
    a. continuing to run even if a part of it is blocked
    b. waiting for one part to finish before the other begins
    c. asking the user to decide the order of multithreading
    d. both a & b
    e. None of these

30. If the kernel is single threaded, then any user level thread performing a blocking system call will
    a. cause the entire process to run along with the other threads
    b. cause the thread to block with the other threads running
    c. cause the entire process to block even if the other threads are available to run
    d. both a & c
    e. None of the above

31. The degree of Multiprogramming is controlled by
    - a. CPU Scheduler
    - b. Context Switching
    - c. Long-term Scheduler
    - d. Medium term Scheduler
    - e. All of Above

32. Process A creates a child process B and a thread C then which of the following statement is false
    - a. A and B have separate heap
    - b. A and C have separate heap
    - c. A and B have shared code
    - d. A and B have shared data segment
    - e. A and B have separate stack

33. What is true about thread?
    - a. Thread switching does not need to interact with operating system.
    - b. All threads can share same set of open files, child processes.
    - c. Multiple threaded processes use fewer resources.
    - d. All of the above
    - e. a & b

34. Which of the following is true about kernel level thread?
    - a. Implementation is by a thread library at the user level.
    - b. Kernel-level threads are slower to create and manage.
    - c. Multi-threaded applications cannot take advantage of multiprocessing.
    - d. Both B and C
    - e. None of the above

35. Which of the following is true about user level thread?
    - a. User level thread is specific to the operating system.
    - b. User-level routines themselves can be multithreaded.
    - c. User-level threads are faster to create and manage.
    - d. All of the above
    - e. None of the above

There are 2 marks for each MCQ in the next 5 questions

36. What output will the following code produce?

```cpp
#include <iostream>
#include <pthread.h>
using namespace std;

void *child_thread(void *arg) {
    int *num = (int*)arg;
    cout <<*num << "." << endl;
    return NULL;
}
```

```
void *parent_thread(void *arg) {
    int arg_child = 1234;
    pthread_t child;
    cout << "thread ";
    pthread_create(&child, NULL, child_thread, &arg_child);
    pthread_join(child, NULL);
    return NULL;
}
int main() {
    pthread_t parent;
    pthread_create(&parent, NULL, parent_thread, NULL);
    pthread_join(parent, NULL);
    return 0;
}
```

ROUGH WORK

a. Code will produce error
b. thread 1234. ✓
c. thread
d. 1234. ✗
e. thread .

37. What will be the size of stack for newly created thread in the following program?

```cpp
#include <iostream>
#include <pthread.h>
using namespace std;

void *thread_func(void *arg) {
    pthread_attr_t *attr = (pthread_attr_t*)arg;
    size_t stacksize;
    pthread_attr_getstacksize(attr, &stacksize);
    return NULL;
}

int main() {
    pthread_t thread;
    pthread_attr_t attr;
    size_t stacksize = 102400;
    pthread_attr_init(&attr);
    pthread_attr_setstacksize(&attr, stacksize);
    pthread_create(&thread, &attr, thread_func, &attr);
    pthread_join(thread, NULL);
    pthread_attr_destroy(&attr);
    return 0;
}
```

$$\frac{102400}{1024} = 100KB$$

ROUGH WORK

a. 102400 Bits
b. 102400 Bytes
c. 102400 KB
d. 102400 MB
e. None of the above

38. What output will the following code produce?

```cpp
#include <iostream>
#include <pthread.h>
using namespace std;

struct ThreadData {
    int arg1;
    int arg2;
    int result;
};
void* threadFunc(void* arg) {
    ThreadData data = *(ThreadData*)(arg);
    data.result = data.arg1 + data.arg2;        2+3
    pthread_exit(&data);
}
int main() {
    pthread_t thread;
    ThreadData data = {2, 3, 0};
    pthread_create(&thread, NULL, threadFunc, &data);
    void* result;
    pthread_join(thread, &result);
    ThreadData* threadResult = static_cast<ThreadData*>(result);
    cout << "Result: " << threadResult->result << endl;
    return 0;
}
```

ROUGH WORK

a. Result: 5
b. Result: 0
c. Result: 2
d. Result: 3
e. -1

CamScanner

39. What output will the following code produce?

```cpp
#include <iostream>
#include <pthread.h>
using namespace std;

struct ThreadArgs {
    pthread_attr_t attr;
    int arg1;
    char arg2;
};
void *thread_func(void *arg) {
    ThreadArgs *args = (ThreadArgs*)arg;
    cout << "Thread started ";
    size_t stacksize;
    pthread_attr_getstacksize(&args->attr, &stacksize);
    cout << args->arg1 << args->arg2 << endl;
    return NULL;
}
int main() {
    pthread_t thread;
    ThreadArgs args;
    args.arg1 = 42;
    args.arg2 = 88;
    size_t stacksize = 102400;
    pthread_attr_init(&args.attr);
    pthread_attr_setstacksize(&args.attr, stacksize);
    pthread_create(&thread, &args.attr, thread_func, &args);
    pthread_join(thread, NULL);
    pthread_attr_destroy(&args.attr);
    return 0;
}
```

ROUGH WORK

a. Thread started 4288
b. Code will result in error
c. Thread started 42X
d. Thread started
e. 4288

40. What output will the following code produce?

```cpp
#include <iostream>
#include <pthread.h>

void* printHello(void* arg) {
    return nullptr;
}

int main() {

    pthread_t thread;
    // Create the thread
    if (pthread_create(&thread, nullptr, printHello, nullptr) != 0) {
        std::cerr << "Error: Failed to create the thread" << std::endl;
        return 1;
    }

    pthread_join(thread, nullptr);
    pthread_join(thread, nullptr);
    std::cout << "Hello from the main thread!" << std::endl;
    return 0;
}
```

If fail

ROUGH WORK

a. Hello from the main thread!
b. Code will result in error
c. Code will end without showing any output
d. All of the above

THE END ☺
ANSWER SHEET ON THE NEXT PAGE

National University of Computer and Emerging Sciences

FAST School of Computing   Spring-2023   **Islamabad Campus**

NOTE: DO NOT DETACH THIS ANSWER SHEET

- Mark your answers to the MCQ's in the following answer sheet by **FILLING** the correct box. **Tick or Cross WILL NOT BE MARKED. You must fill the whole box.**
- Any answers not filled here **WILL NOT BE MARKED.** Similarly, **Multiple filled choices will be marked incorrect.**
- Filling the ROLL NO. is a MUST. NEGATIVE MARKING (-2) if you don't fill it.

NAME: _____

Operating Systems

Section 1

Section 2

18

06

Zero