DIGITAL SIGNAL PROCESSING LAB (CSE-3222)

1.Generating elementary signals like Unit Step, Ramp, Exponential, Sine, and Cosine sequences.

2.Demonstrates the effect of sampling, aliasing.

3.Show that the highest rate of oscillation in a discrete-time sinusoidal is obtained when ω=π.

4.Consider the continuous-time analog signal x(t)=3cos(100πt). Sample the analog signal at 200 Hz and 75 Hz. Show the discrete-time signal after sampling. ⇒ realization.

5. Generate and plot a triangular sequence and a rectangular pulse sequence. Compare their properties in time domain.

6. Consider a continuous-time analog signal $x(t) = 2sin(50\pi t) + 4cos(150\pi t)$.

Microprocessor and Assembly Language LAB (CSE-3232)

Using the STM32F103C8T6 microcontroller and an ST-Link v2 programmer, write a Blinky program where the LED blinks at different time intervals:

1. The LED should blink every 1 second.
2. The LED should blink every 3 seconds.
3. The LED should blink every 2 seconds.

Using STM32CubeMX and System Workbench for STM32 (SW4STM32):

→ Configure pin PC13 as a GPIO Output.
→ Write the LED on/off logic inside the while(1) loop.
→ Implement different delays (1s, 3s, 2s) to demonstrate the blinking at different intervals.

Computer Networks LAB (CSE-3252)

1. Building Ethernet cable-straight and crossover cable (for both T568A & T568B standards). 2. Assigning and checking IP configuration of a machine. 3. Checking connectivity in between two nodes (using ping command). 4. Windows file/folder/printer sharing.
5. Cisco packet tracer - simulating HUB & Switch operation. 6. Cisco packet tracer - Creating VLAN in cisco switch. 7. Cisco packet tracer - Basic router configuration. 8. Cisco packet tracer - Inter VLAN routing.
9. Configuring POP3 email server for windows.

Operating System LAB (CSE-3242)

1. Write a C program where the main process (called parent_process) creates 3 child processes.
   - The parent should not create any grandchildren processes (only one level of children).
   - Each child process should perform a different operation (addition, subtraction, multiplication) on two variables initialized in the parent process.
   - Finally, trace and display the parent and child processes in the process tree to show their relationships.

2. Write a C program to create an orphan process.

3. Write a C program to create a zombie process.

4.Write a C program to analyze the effect of local and global variables on a parent process and a child process.

5. Write a C program to show how two related processes can communicate with each other by an unnamed pipe.

6. Write a C program to show how two unrelated processes can communicate with each other by a named pipe.

7. Write a multithreaded C program that calculates three statistical results from a list of numbers. The program will receive a sequence of integers from the command line and then create three worker threads:
   1. One thread should compute the average of the numbers.
   2. Another thread should compute the maximum value.
   3. A third thread should compute the minimum value.

The results (average, minimum, maximum) must be stored in global variables. Once all worker threads have finished, the main thread will display the computed results.
Example:
If the program is given the integers:
45 67 89 23 56 91 38
The program should print:
A. The average value is 58
B. The minimum value is 23
C. The maximum value is 91

8. Write a C program to create a multi-threaded process and investigate the following cases:

A. When one thread in the process calls fork(), does the newly created process contain all the threads of the parent, or is it single-threaded?

B. When a thread executes the exec() system call, does it replace the entire code of the process, or only that thread?

C. If exec() is called immediately after a fork(), will the child process still contain all the parent's threads, or will it only duplicate the calling thread?