Question 10 [6 Points]

Write a C program demonstrating signal handling.

• Parent process runs indefinitely

• Child process sends SIGTERM after 5 seconds

• Another child sends SIGINT after 10 seconds

• Parent handles each signal differently and exits gracefully

-----

Script:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <sys/types.h>

// Handler for SIGTERM
void handle_sigterm(int sig) {
    printf("\nParent received SIGTERM (signal %d).\n", sig);
    printf("Handling SIGTERM: Performing cleanup...\n");
    printf("Parent exiting gracefully after SIGTERM.\n");
    exit(0);
}

// Handler for SIGINT
void handle_sigint(int sig) {
    printf("\nParent received SIGINT (signal %d).\n", sig);
    printf("Handling SIGINT: Saving state...\n");
    printf("Parent exiting gracefully after SIGINT.\n");
    exit(0);
}

int main() {
    pid_t pid1, pid2;

    // Register signal handlers
    signal(SIGTERM, handle_sigterm);
    signal(SIGINT, handle_sigint);

    printf("Parent process started. PID = %d\n", getpid());
    printf("Parent running indefinitely...\n");

    // First child: sends SIGTERM after 5 seconds
    pid1 = fork();
    if (pid1 == 0) {
        sleep(5);
        printf("\nChild 1 sending SIGTERM to parent.\n");
        kill(getppid(), SIGTERM);
        exit(0);
    }

    // Second child: sends SIGINT after 10 seconds
    pid2 = fork();
    if (pid2 == 0) {
        sleep(10);
        printf("\nChild 2 sending SIGINT to parent.\n");
        kill(getppid(), SIGINT);
        exit(0);
    }

    // Parent runs indefinitely
    while (1) {
        pause();   // Wait for signals
    }

    return 0;
}
```

⌘L to chat, ⌘K to generate

Outcome

```
● sudhanshu@Sudhanshus-MacVati Question9 % cd ..
● sudhanshu@Sudhanshus-MacVati Graded Lab Assignment — (Modules 5—10) % cd Question10
● sudhanshu@Sudhanshus-MacVati Question10 % ls
  signal_demo.c
● sudhanshu@Sudhanshus-MacVati Question10 % gcc signal_demo.c —o signal_demo
● sudhanshu@Sudhanshus-MacVati Question10 % ./signal_demo

  Parent process started. PID = 6161
  Parent running indefinitely...

  Child 1 sending SIGTERM to parent.

  Parent received SIGTERM (signal 15).
  Handling SIGTERM: Performing cleanup...
  Parent exiting gracefully after SIGTERM.
○ sudhanshu@Sudhanshus-MacVati Question10 %
  Child 2 sending SIGINT to parent.
```