```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>
```

#include <fcntl.h> \\ provides the definitions for file control options, file descriptor flags, and related functions (open, creat, fcntl).

```
#include <sys/file.h>
```

\\ header provides declarations for file locking functions like flock, which allows processes to apply shared or exclusive locks to files to prevent conflicts during file access.

```
#include <time.h>
```

\\provides functions and macros for working with time and date. It includes functionality to manipulate, format, and retrieve system time.

```
#include <string.h>

#include <sys/wait.h>
```

Q1 :

```
void cpu_only_task() {
 long count = 0;
 for (long i = 0; i < 1e8; i++) {
    count += i;  // Simple CPU computation
 }
}
```

```
// Helper function to calculate elapsed time
double get_elapsed_time(struct timespec start, struct timespec end) {
return (end.tv_sec - start.tv_sec) + (end.tv_nsec - start.tv_nsec) / 1e9;
}
```

Q2

```
#define DATA "Hello word.\n"
void cpu_io_task() {
int f;
int fd = open("io_demo.txt", O_CREAT | O_WRONLY | O_TRUNC, 0666);
 if (fd < 0) {
    perror("Error opening file for CPU and I/O task");
    exit(EXIT_FAILURE);
 }
 for (int i = 0; i < 100; i++) {
    long count = 0;
    for (long j = 0; j < 1e6; j++) {
       count += j;
    }
 f= write(fd, DATA, strlen(DATA));
    if (f < 0) {
       perror("Write failed");
       close(fd);
       exit(EXIT_FAILURE);
    }
    fsync(fd);  // Ensure data is written to disk
 }
```