Operating System
Lab 3

Question 1

```c
#include <stdio.h>
#include <unistd.h>

int main() {

    pid_t pid = getpid();
    printf("Process ID after execv system call in
my_info.c: %d\n", pid);

    printf("Name: Areeba\n");
    return 0;
}
```

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
```

```c
int main() {

    pid_t pid = getpid();
    printf("Process ID before calling the execv
system call execv: %d\n", pid);

    char *Path = "./my_info";

    char *args[] = {Path, NULL};
    if (execv(Path, args) == -1) {
        perror("execv");
        exit(EXIT_FAILURE);
    }

    return 0;
}
```

Question 2

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

int main() {
    pid_t pid1, pid2, pid3, pid4;
    int status;
```

```
    pid1 = fork();
    if (pid1 == 0) {
        execlp("mkdir", "mkdir", "demo_folder",
NULL);
        perror("Error executing mkdir");
        _exit(1);
    } else {
        wait(NULL);
    }
    pid2 = fork();
    if (pid2 == 0) {
        char* args[] = {"touch",
"demo_folder/file1.txt", "demo_folder/file2.txt",
NULL};
        execvp("touch", args);
        perror("Error executing touch");
        _exit(1);
    } else {
        wait(NULL);
    }
    pid3 = fork();
    if (pid3 == 0) {
        char* args[] = {"ls", "demo_folder", NULL};
        execvp("ls", args);
        perror("Error executing ls");
        _exit(1);
    } else {
        wait(NULL);
    }
```

```
    pid4 = fork();
    if (pid4 == 0) {
        char* args[] = {"rm", "-rf", "demo_folder",
NULL};
        execvp("rm", args);
        perror("Error executing rm");
        _exit(1);
    } else {
        wait(NULL);
    }

    return 0;
}
```

```
~/lab3prep$ gcc main.c -o main
~/lab3prep$ ./main
file1.txt  file2.txt
~/lab3prep$ gcc main.c -o main
~/lab3prep$ ./main
file1.txt  file2.txt
~/lab3prep$ gcc main.c -o main
~/lab3prep$ ./main
mkdir: cannot create directory ‘demo_folder’: File exists
file1.txt  file2.txt
~/lab3prep$ gcc main.c -o main
~/lab3prep$ ./main
file1.txt  file2.txt
~/lab3prep$
```

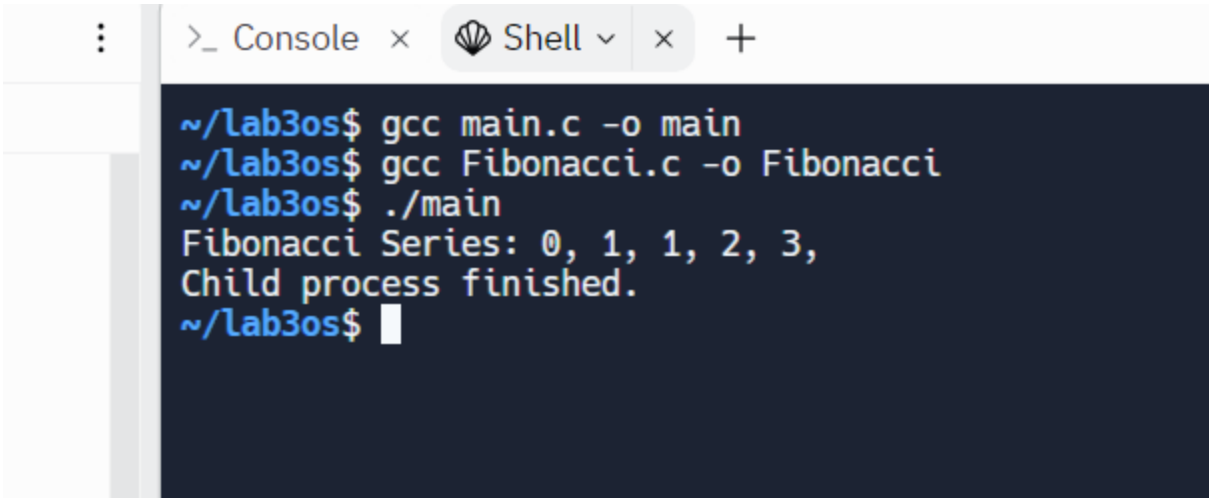
Question 3

```c
#include <stdio.h>
#include <unistd.h>
```

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

int main() {
    pid_t pid = fork();

    if (pid == 0) {
        // Child process
        char *args[] = {"Fibonacci", "5", NULL};
        execv("./Fibonacci", args);

        perror("execv");
        return 1;
    } else if (pid > 0) {
        wait(NULL);
        printf("Child process finished.\n");
    } else {
        perror("fork");
        return 1;
    }

    return 0;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
```

```c
void fibonacci(int n) {
    int i, t1 = 0, t2 = 1, next;
    printf("Fibonacci Series: ");
    for (i = 1; i <= n; i++) {
        printf("%d, ", t1);
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
    }
    printf("\n");
}
int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("\n", argv[0]);
        return 1;
    }
    int n = atoi(argv[1]);
    fibonacci(n);
    return 0;
}
```


Console
Shell
~/lab3os$ gcc main.c -o main
~/lab3os$ gcc Fibonacci.c -o Fibonacci
~/lab3os$ ./main
Fibonacci Series: 0, 1, 1, 2, 3,
Child process finished.
~/lab3os$