B. Siva Shirish -192324016

1. **Create a new process by invoking the appropriate system call. Get the process identifier of the currently running process and its respective parent using system calls and display the same using a C program.**

Aim:

To create a new process using the fork() system call, retrieve the process identifier (PID) of the current process and its parent process, and display them.

Algorithm:

1. Start the program.
2. Use the fork() system call to create a new process.
   - fork() returns:
     - 0 for the child process.
     - A positive PID for the parent process.
   - A negative value indicates failure.
3. In the child process:
   - Retrieve the PID using getpid().
   - Retrieve the parent PID using getppid().
   - Display the details.
4. In the parent process:
   - Retrieve the PID using getpid().
   - Retrieve the parent PID using getppid().
   - Display the details.
5. End the program.

Procedure:

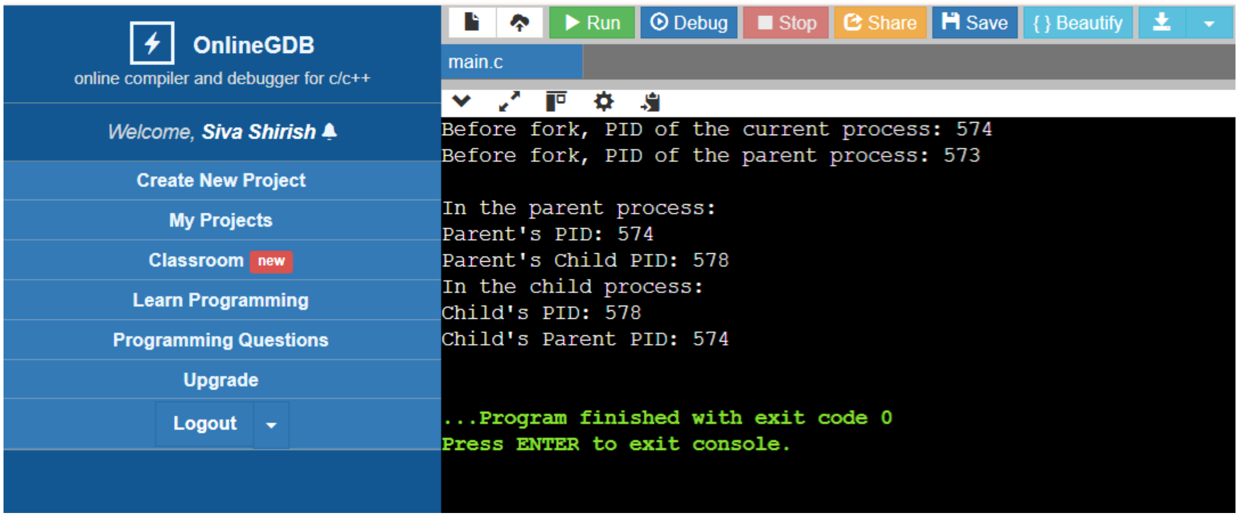
1. Include the necessary headers: <stdio.h> and <unistd.h>.
2. Use the fork() function to create a new process.
3. Use getpid() and getppid() to get the PID and parent PID.
4. Differentiate behavior for child and parent processes using the return value of fork().

5. Print the information to the console.

CODE:

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

int main() {
    pid_t pid;
    printf("Before fork, PID of the current process: %d\n", getpid());
    printf("Before fork, PID of the parent process: %d\n\n", getppid());

    pid = fork();

    if (pid == -1) {

        perror("Fork failed");
        return 1;
    } else if (pid == 0) {

        printf("In the child process:\n");
        printf("Child's PID: %d\n", getpid());
        printf("Child's Parent PID: %d\n", getppid());
    } else {
        printf("In the parent process:\n");
        printf("Parent's PID: %d\n", getpid());
        printf("Parent's Child PID: %d\n", pid);
    }

    return 0;
}
```

Output:



```
Before fork, PID of the current process: 574
Before fork, PID of the parent process: 573

In the parent process:
Parent's PID: 574
Parent's Child PID: 578
In the child process:
Child's PID: 578
Child's Parent PID: 574


...Program finished with exit code 0
Press ENTER to exit console.
```