

```
1] #include <stdio.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <unistd.h>
#include <unistd.h>
int main(void)
{
    int exitstatus;
    int exitval = 10;
    pid_t pid;
    pid = fork();
    if (pid == 0)
    {
        printf("in child process\n");
        printf("CHILD: terminating with exit value %d\n", exitval);
        exit(exitval);
    }
    else
    {
        wait(&exitstatus);
        printf("In Parent Process\n");
        printf("Parent child terminated with exit value %d\n",
               WEXITSTATUS(exitstatus));
        exit(20);
    }
    printf("Both Processes continue from here\n\n");
    exit(0);
}
```

2]

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

int main()
{
    int Pid;
    printf("Before calling fork: \n");
    Pid = fork();
    if (Pid > 0)
    {
        sleep(3);
        printf("In Parent Process Id is : %d \n Parents parent process id  
is : %d \n (child process id is : %d \n)", getpid(), getppid(),  
Pid);
    }
    else if (Pid == 0)
    {
        printf("Child process id is : %d \n Parent process id is : %d",  
getpid(), getppid());
    }
    else
    {
        printf("An error occurred while calling fork()");
        exit(1);
    }
    printf("In Both processes continues here \n");
    exit(0);
}
```