# Practical 5

**Aim :** [B] In continuation of part (a), write a C program where the parent process waits for the child process to terminate.

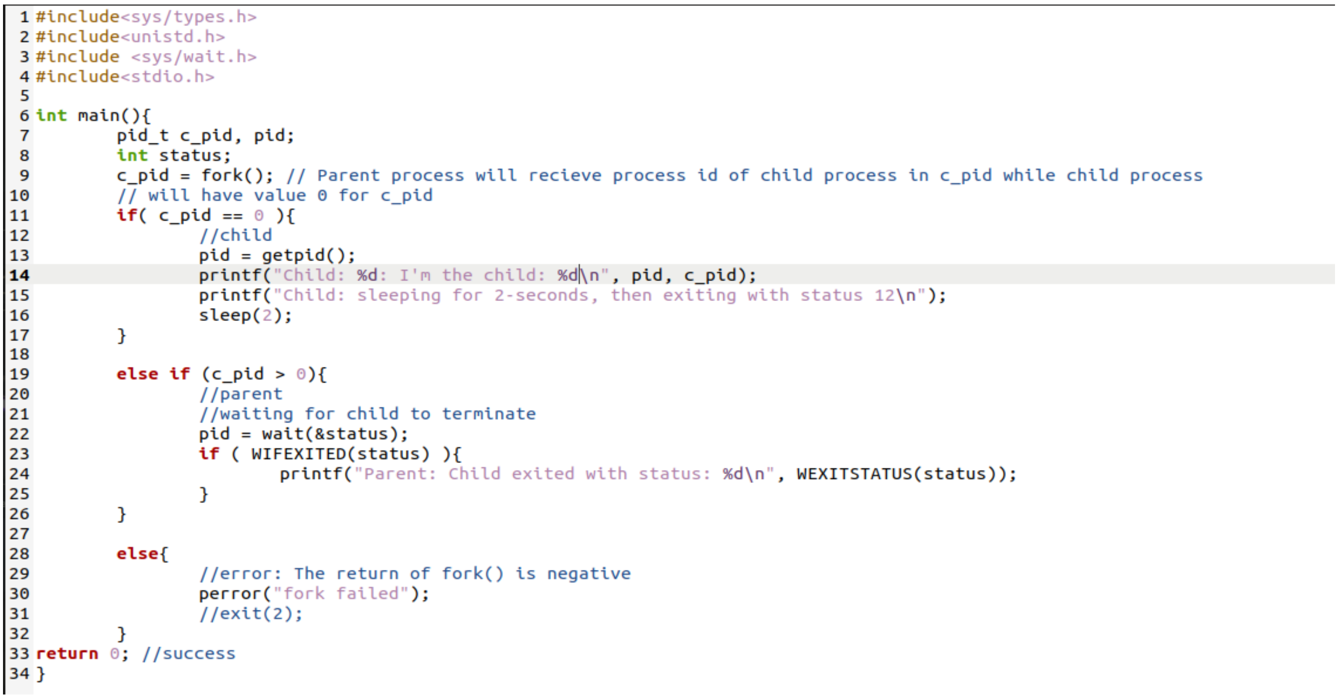
**Program Code :**

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

int main(){
        pid_t c_pid, pid;
        int status;
        c_pid = fork(); // Parent process will recieve process id of child process in c_pid
while child process
        // will have value 0 for c_pid
        if( c_pid == 0 ){
                //child
                pid = getpid();
                printf("Child: %d: I'm the child: %d\n", pid, c_pid);
                printf("Child: sleeping for 2-seconds, then exiting with status 12\n");
                sleep(2);
        }

        else if (c_pid > 0){
                //parent
                //waiting for child to terminate
                pid = wait(&status);
                if ( WIFEXITED(status) ){
                        printf("Parent: Child exited with status: %d\n",
WEXITSTATUS(status));
                }
        }

        else{
                //error: The return of fork() is negative
                perror("fork failed");
                //exit(2);
        }
return 0; //success
}
```



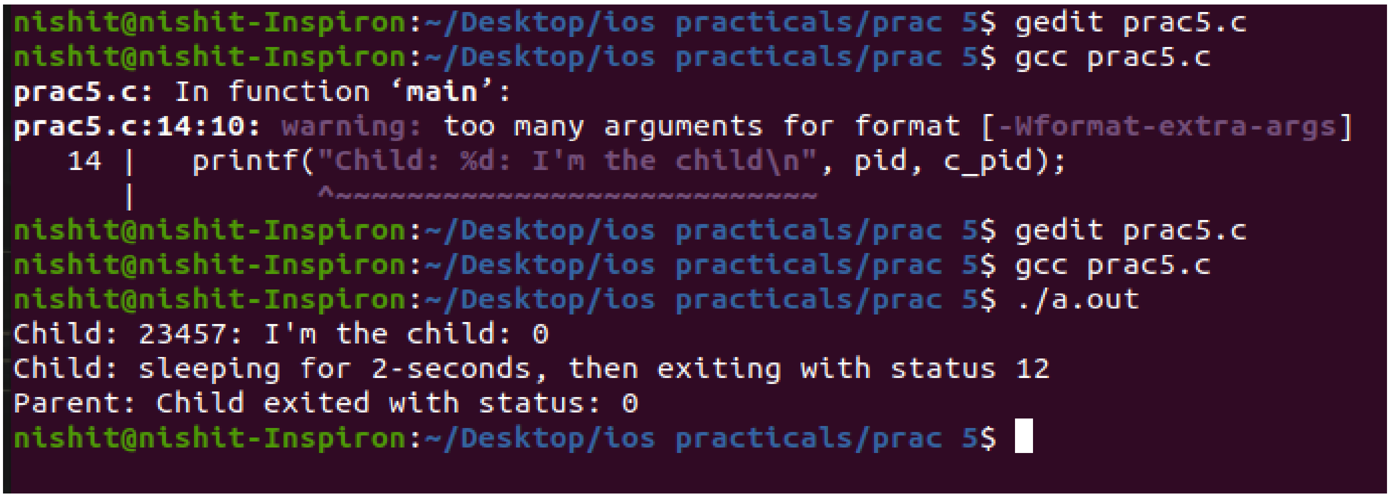
```c
#include<sys/types.h>
#include<unistd.h>
#include <sys/wait.h>
#include<stdio.h>

int main(){
        pid_t c_pid, pid;
        int status;
        c_pid = fork(); // Parent process will recieve process id of child process in c_pid while child process
        // will have value 0 for c_pid
        if( c_pid == 0 ){
                //child
                pid = getpid();
                printf("Child: %d: I'm the child: %d\n", pid, c_pid);
                printf("Child: sleeping for 2-seconds, then exiting with status 12\n");
                sleep(2);
        }

        else if (c_pid > 0){
                //parent
                //waiting for child to terminate
                pid = wait(&status);
                if ( WIFEXITED(status) ){
                        printf("Parent: Child exited with status: %d\n", WEXITSTATUS(status));
                }
        }

        else{
                //error: The return of fork() is negative
                perror("fork failed");
                //exit(2);
        }
return 0; //success
}
```

**Output :**



```
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gedit prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gcc prac5.c
prac5.c: In function ‘main’:
prac5.c:14:10: warning: too many arguments for format [-Wformat-extra-args]
   14 |    printf("Child: %d: I'm the child\n", pid, c_pid);
      |          ^~~~~~~~~~~~~~~~~~~~~~~~~~~
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gedit prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gcc prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ ./a.out
Child: 23457: I'm the child: 0
Child: sleeping for 2-seconds, then exiting with status 12
Parent: Child exited with status: 0
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ 
```