

```
/*distinguish parent from child process*/
```

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

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

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

```
#define MAX_COUNT 5
```

```
void ChildProcess();      /* child process prototype */
```

```
void ParentProcess();     /* parent process prototype */
```

```
void main()
```

```
{
```

```
    pid_t pid;
```

```
    pid = fork();
```

```
    if (pid == 0)
```

```
        ChildProcess();
```

```
    else
```

```
        ParentProcess();
```

```
}
```

```
void ChildProcess()
```

```
{
```

```
    int i;
```

```
    for (i = 1; i <= MAX_COUNT; i++)
```

```
        printf(" This line is from child, value = %d\n", i);
```

```
    printf(" *** Child process is done ***\n");
```

```
}
```

```
void ParentProcess()
```

```
{
```

```
    int i;
```

```

for (i = 1; i <= MAX_COUNT; i++)

    printf("This line is from parent, value = %d\n", i);

printf("*** Parent is done ***\n");
}

```

The screenshot shows the Programiz Online C Compiler interface. The code editor contains a C program named `main.c` that uses `fork()` to create a child process. The parent process prints five lines of output, followed by the child process printing five lines of output. The output window on the right shows the execution results, confirming that the parent process completed its execution before the child process began.

```

main.c
10  pid_t pid;
11
12  pid = fork();
13  if (pid == 0)
14      ChildProcess();
15  else
16      ParentProcess();
17 }
18
19 void ChildProcess()
20 {
21     int i;
22
23     for (i = 1; i <= MAX_COUNT; i++)
24         printf("This line is from child, value = %d\n", i);
25     printf(" *** Child process is done ***\n");
26 }
27
28 void ParentProcess()
29 {
30     int i;
31
32     for (i = 1; i <= MAX_COUNT; i++)
33         printf("This line is from parent, value = %d\n", i);
34     printf("*** Parent is done ***\n");
35 }

```

Output

```

This line is from parent, value = 1
This line is from parent, value = 2
This line is from parent, value = 3
This line is from parent, value = 4
This line is from parent, value = 5
*** Parent is done ***
This line is from child, value = 1
This line is from child, value = 2
This line is from child, value = 3
This line is from child, value = 4
This line is from child, value = 5
*** Child process is done ***

```