

/*distinguish parent from child process*/

#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

#define MAX_COUNT 5

void child_process();

void parent_process();

void main(){

pid_t f_val;

f_val=fork();

if(f_val==0){

child_process();

}

else{

parent_process();

}

}

void child_process(){

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

printf("this line is from child, value:%d\n",i);

}

printf("child process ends**\n");**

}

void parent_process(){

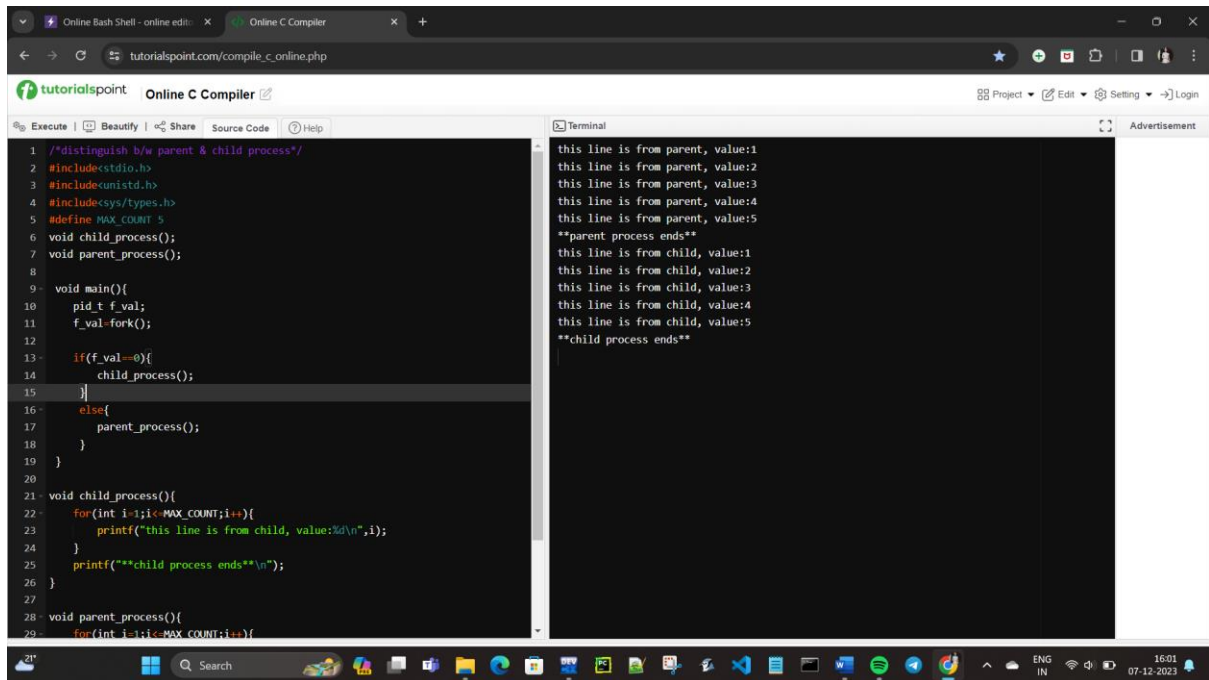
for(int i=1;i<=MAX_COUNT;i++){

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

}

```
printf("**parent process ends**\n");
```

```
}
```



The screenshot shows the 'Online C Compiler' interface on the website 'tutorialspoint.com'. The code editor contains a C program that uses `fork()` to create a child process. The parent process prints five lines of output, followed by `**parent process ends**`. The child process prints five lines of output, followed by `**child process ends**`. The terminal window on the right displays the output of the program, showing the parent's output first, followed by the child's output.

```
1  /*distinguish b/w parent & child process*/
2  #include<stdio.h>
3  #include<unistd.h>
4  #include<sys/types.h>
5  #define MAX_COUNT 5
6  void child_process();
7  void parent_process();
8
9  void main(){
10     pid_t f_val;
11     f_val=fork();
12
13     if(f_val==0){
14         child_process();
15     }
16     else{
17         parent_process();
18     }
19 }
20
21 void child_process(){
22     for(int i=1;i<=MAX_COUNT;i++){
23         printf("this line is from child, value:%d\n",i);
24     }
25     printf("**child process ends**\n");
26 }
27
28 void parent_process(){
29     for(int i=1;i<=MAX_COUNT;i++){
```

Terminal 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 process ends**
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 ends**
```