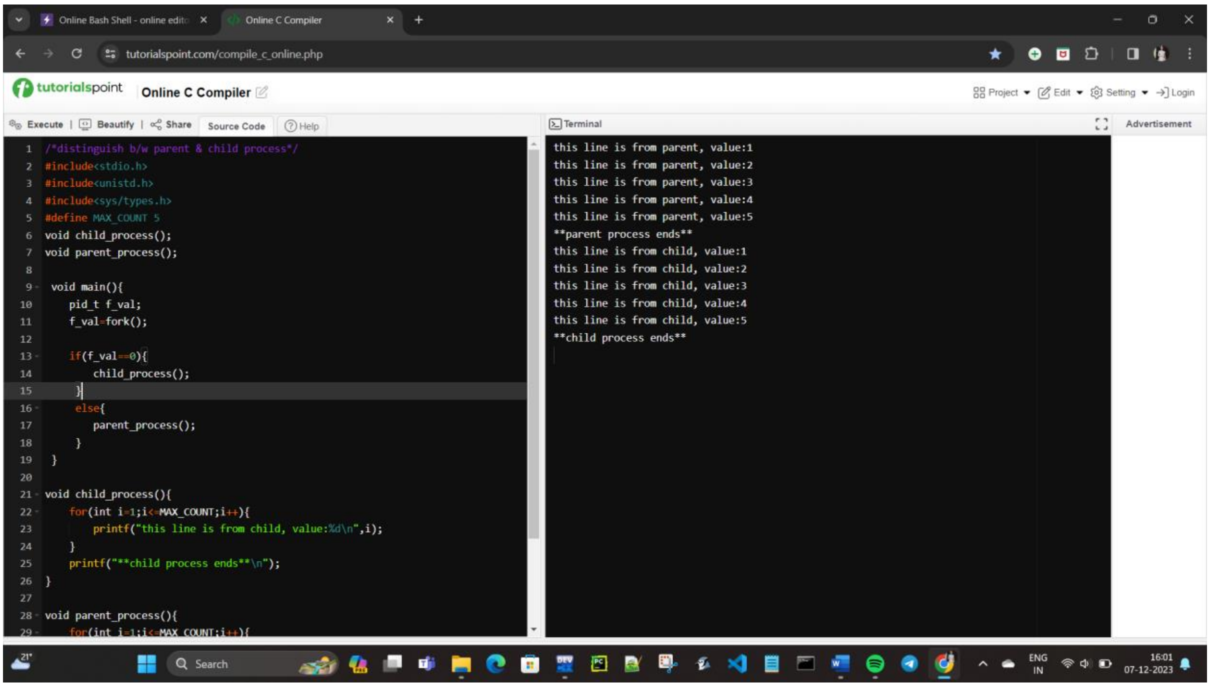
/*distinguish parent from child process*/

```
#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");
}
```



```
/*distinguish b/w parent & 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++){
```

```
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**
```