Experiment 9: - Write a program in C that creates a child process, waits for the termination of the child and lists its PID.

Syntax :

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/wait.h>
int main(){
  pid_t p;
printf("Before fork\n");
p =fork();
if(p==0){
  printf("I am child having id: %d\n",getpid());
  printf("My parent's id: %d\n",getppid());
 }else{
 Wait(NULL);
printf("My child's is : %d\n",p);
printf("I am parent having id : %d\n", getpid());
 }
printf("Done\n");
return 0;
}
```

```
┌──(aakash㉿kali)-[~/Desktop]
└─$ nano exp8.c

┌──(aakash㉿kali)-[~/Desktop]
└─$ gcc exp8.c

┌──(aakash㉿kali)-[~/Desktop]
└─$ ./a.out
before fork
I am child having pid:8309
My parent pid: 8308
common
my child  id: 8309
I am parent having  pid: 8308
common
```

```c
#include<unistd.h>
#include<sys/types.h>
#include<stdio.h>
#include<sys/wait.h>
int main()
{
pid_t p;
printf("before fork\n");
p=fork();
if(p==0)
{
printf("I am child having pid:%d \n",getpid());
printf("My parent pid: %d \n",getppid());
}
else
{
wait(NULL);
printf("my child  id: %d\n",p);
printf("I am parent having  pid: %d\n",getpid());
}
printf("common\n");
}
```