EXPERIMENT 9:

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

```
#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) // child
{
printf("I am child having id %d\n",getpid());
printf("My parent id is %d\n",getpid());
}
else //parent
{
wait(NULL);
printf("My child id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
printf("Common\n");
}
```

CODE:

```
  GNU nano 6.2
#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) // child
{
printf("I am child having id %d\n",getpid());
printf("My parent id is %d\n",getpid());
}
else //parent
{
wait(NULL);
printf("My child id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
printf("Common\n");
}
```

OUTPUT:

```
suraj@surajpandit:~$ gcc fork.c
suraj@surajpandit:~$ ./a.out
before fork
I am child having id 11810
My parent id is 11810
Common
My child id is 11810
I am parent having id 11809
Common
suraj@surajpandit:~$
```