Write a C program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen.

Program:

```
#include <stdio.h>
#include <sys/wait.h>  /* contains prototype for wait */
int main(void)
{
int pid;
int status;
printf("Hello World!\n");
pid = fork( );
if(pid == -1) /* check for error in fork */
{
perror("bad fork");
exit(1);
}
if (pid == 0)
printf("I am the child process.\n");
else
{
wait(&status); /* parent waits for child to finish */
printf("I am the parent process.\n");
}
}
```

Output:

```
student@ubutnu:$gcc –o child.out child.c
student@ubutnu: ./child.out
Hello World!
I am the child process.
I am the parent process
```

Write a C program in which a parent writes a message to a pipe and the child reads the message

#create a file name as w6.c

#then write program below

```
#include<stdio.h>
#include<unistd.h>
int main() {
  int pipefds1[2], pipefds2[2];
  int returnstatus1, returnstatus2; int
  pid;
```

```
  char pipe1writemessage[20] = "Hi"; char
  pipe2writemessage[20] = "Hello"; char
  readmessage[20];
  returnstatus1 = pipe(pipefds1);
  if (returnstatus1 == -1) { printf("Unable
    to create pipe 1 \n"); return 1;
  }
  returnstatus2 = pipe(pipefds2);

  if (returnstatus2 == -1) { printf("Unable
    to create pipe 2 \n"); return 1;
  }
  pid = fork();

  if (pid != 0)
{
    close(pipefds1[0]);
    close(pipefds2[1]);
    printf("In Parent: Writing to pipe 1 – Message is %s\n", pipe1writemessage);
    write(pipefds1[1], pipe1writemessage, sizeof(pipe1writemessage));
    read(pipefds2[0], readmessage, sizeof(readmessage));
    printf("In Parent: Reading from pipe 2 – Message is %s\n", readmessage);
  }
else
{
    close(pipefds1[1]);
    close(pipefds2[0]);
```

```
    read(pipefds1[0], readmessage, sizeof(readmessage));
    printf("In Child: Reading from pipe 1 – Message is %s\n", readmessage); printf("In
    Child: Writing to pipe 2 – Message is %s\n", pipe2writemessage); write(pipefds2[1],
    pipe2writemessage, sizeof(pipe2writemessage));
  }
  return 0;
}
#then create a file name as w6.sh
#then write below statements
-------------------------------------------------------------
gcc w6.c
./a.out w6.c
-------------------------------------------------------------
#then save the files and execute
```