1(a)

Open *q1.c ~/

```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
int main(){
pid_t C1 = fork();
if(C1 ==0)
{
fprintf(stderr,"Child Process id: %d",getpid());
while(1);
}
else
{
fprintf(stderr,"Parent Process id: %d",getpid());
while(1);
}
}
```

```
root@DESKTOP-8EIFV6T:~# gcc q1.c
root@DESKTOP-8EIFV6T:~# ./a.out
Parent Process id: 8226Child Process id: 8227
```

```
root@DESKTOP-8EIFV6T:~# ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 R     0    8226     354 99  80   0 -   661 -      pts/0    00:00:21 a.out
1 R     0    8227    8226 99  80   0 -   661 -      pts/0    00:00:21 a.out
4 R     0    8311    5528  0  80   0 -  1871 -      pts/2    00:00:00 ps
root@DESKTOP-8EIFV6T:~#
```

1(b)

```
root@DESKTOP-8EIFV6T:~# kill -9 8227
root@DESKTOP-8EIFV6T:~# ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 R     0    8226     354 99  80   0 -   661 -      pts/0    00:01:03 a.out
1 Z     0    8227    8226 88  80   0 -     0 -      pts/0    00:00:56 a.out <defunct>
4 R     0    8480    5528  0  80   0 -  1871 -      pts/2    00:00:00 ps
root@DESKTOP-8EIFV6T:~#
```

1(c)

```
root@DESKTOP-8EIFV6T:~# gcc q1.c
root@DESKTOP-8EIFV6T:~# ./a.out
Parent Process id: 8917Child Process id: 8918
```

```
root@DESKTOP-8EIFV6T:~# kill -9 8917
root@DESKTOP-8EIFV6T:~# ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 R     0    8918     352 99  80   0 -   661 -      pts/0    00:00:39 a.out
4 R     0    9077    5528  0  80   0 -  1871 -      pts/2    00:00:00 ps
root@DESKTOP-8EIFV6T:~#
```

1(d)

Open *q1.c ~/

```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
int main(){
pid_t C1 = fork();
if(C1 ==0)
{
fprintf(stderr,"Child Process id: %d",getpid());
while(1);
}
else
{
fprintf(stderr,"Parent Process id: %d",getpid());
wait(NULL);
while(1);
}
}
```

```
root@DESKTOP-8EIFV6T:~# gcc q1.c
root@DESKTOP-8EIFV6T:~# ./a.out
Parent Process id: 11533Child Process id: 11534
```

```
root@DESKTOP-8EIFV6T:~# ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S     0   11533     354  0  80   0 -   661 do_wai pts/0    00:00:00 a.out
1 R     0   11534   11533 99  80   0 -   661 -      pts/0    00:01:08 a.out
4 R     0   11803    5528  0  80   0 -  1871 -      pts/2    00:00:00 ps
```

1(e)

```
root@DESKTOP-8EIFV6T:~# kill -9 11534
root@DESKTOP-8EIFV6T:~# ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 R     0   11533     354  1  80   0 -   661 -      pts/0    00:00:01 a.out
4 R     0   12096    5528  0  80   0 -  1871 -      pts/2    00:00:00 ps
root@DESKTOP-8EIFV6T:~#
```

3

Open q3.c Save

```c
#include<stdio.h>
#include<sys/wait.h>
#include<unistd.h>
int main(){
 if(vfork()==0)
 {
  fprintf(stderr,"\nChild process id :%d  Parent process id :%d\n",getpid(),getppid());
  execl("/usr/bin/cp","cp","file1","file2",NULL);
  _exit(0);
 }
  sleep(1);
  if(vfork()==0)
 {
  fprintf(stderr,"\nChild process id :%d  Parent process id :%d\n",getpid(),getppid());
  execl("/usr/bin/cat","cat","file2",NULL);
  _exit(0);
  }
  sleep(1);
  if(vfork()==0)
 {
  fprintf(stderr,"\nChild process id :%d  Parent process id :%d\n",getpid(),getppid());
  execl("/usr/bin/sort","sort","-r","file2",NULL);
  _exit(0);
  }

   sleep(1);
   fprintf(stderr,"\nParent process id :%d",getpid());

}
```

```
┌──(munu㉿kali)-[~]
└─$ cat file1
1
2
3
4
5
6

┌──(munu㉿kali)-[~]
└─$ cat file2
1
2
3
4
5
6
```

```
┌──(munu㉿kali)-[~]
└─$ gedit q3.c
^C
┌──(munu㉿kali)-[~]
└─$ gcc q3.c
┌──(munu㉿kali)-[~]
└─$ ./a.out
Child process id :7411  Parent process id :7410

Child process id :7420  Parent process id :7410
1
2
3
4
5
6

Child process id :7429  Parent process id :7410
6
5
4
3
2
1

Parent process id :7410
```

4

Open q4.c

```c
#include<stdio.h>
#include<unistd.h>
#include<stdbool.h>
int main() {
    int length=10;
    printf("Enter the length : ");
    scanf("%d", &length);
    int fibArray[length];
    if(vfork()==0){
        fibArray[0] = 0;
        fibArray[1] = 1;
        for (int i = 2; i < length; i++) {
            fibArray[i]=fibArray[i-1]+fibArray[i-2];
        }
        _exit(0);
    }
      printf("\n");
        printf("Fibonacci Series:\n");
        for (int i = 0; i < length; i++) {
                printf("%d ", fibArray[i]);
        }
        printf("\n");
        printf("Prime Fibonacci numbers:\n");
        for (int i = 3; i < length; i++) {
        int isPrime = 1;
        for (int j = 2; j * j <= fibArray[i]; j++) {
        if (fibArray[i] % j == 0)
            isPrime = 0;
        }
        if (isPrime)
            printf("%d at index : %d \n", fibArray[i],i);
    }
    return 0;
}
```

```
┌──(munu㉿kali)-[~]
└─$ gcc q4.c

┌──(munu㉿kali)-[~]
└─$ ./a.out
Enter the length : 10

Fibonacci Series:
0 1 1 2 3 5 8 13 21 34
Prime Fibonacci numbers:
2 at index : 3
3 at index : 4
5 at index : 5
13 at index : 7
```