Theory:

Fork system call is used for creating a new child process. It runs simultaneously with parent process. After a new child process is created, both processes will execute the next instruction following the fork() system call.

fork() takes no parameters and returns an integer value.

Negative Value: creation of a child process was unsuccessful. Zero: Returned to the newly created child process.

Positive value: Returned to parent. The value contains process ID of newly created

child process.

Suppose,

P0

fork()

P0

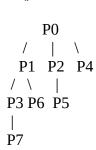
| P1

fork()

P0 | \ P1 P2 |

fork()

Р3



Problem No: 1

Problem Title: If fork is parent then find the sum of even numbers of the array. If fork is child then find the sum of odd numbers of the array.

Code:

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
int main()
int arr[5]={10,11,12,13,14};
int ParentEvenSum=0;
int ChildOddSum=0;
int n=fork();
// n>0 denotes parent fork
if(n>0){
for(int i=0; i<5; i++){
 if(arr[i]%2==0){
  ParentEvenSum+=arr[i];
}
// n=0 denotes child fork
if(n==0)
for(int i=0;i<5;i++){}
 if(arr[i]%2==1){
  ChildOddSum+=arr[i];
}
}
}
if(n<0){
printf("Child process is not created successfully");
printf("Parent Even Sum");
printf("%d",ParentEvenSum);
printf("Child Odd Sum");
printf("%d",ChildOddSum);
return 0;
```

```
Into_variable.sh Subtraction.sh Templates Wideos Showmik@showmik-virtual-machine:~$ gcc ParentChildEvenOdd.c -o PCEO Showmik@showmik-virtual-machine:~$ ./PCEO Parent Even Sum36Child Odd Sum0Parent Even Sum0Child Odd Sum24showmik@showmik-virshowmik@showshshowms showmik@showmik-virtual-machine:~$ ./PCEO Parent Even Sum0Child Odd Sum24showmik@showmik-virshowmik@showshshowms showmik@showmik-virtual-machine:~$ ...

Problem No: 2

Code:

#include<stdio.h>
#include<sys/types.h>
#include<unistd h>
```

```
#include<unistd.h>
int main()
     int a=\{2,3,5,6,9,7,1,2\};
     for(int i = 0; i < 8; i++){
          scanf("%d", a[i]);
     int n1=fork();
     int n2=fork();
     int sum=0;
     if(n1>0\&\&n2>0){
          printf("sum of first two elements= %d\n",a[0]+a[1]);
     else if(n1>0\&\&n2==0){
          printf("sum Of second two elements= %d\n",a[2]-a[3]);
     }
     else if(n1==0\&\&n2>0){
          printf("sum Of third two elements= %d\n",a[4]*a[5]);
     else if(n1==0\&\&n2==0){
          printf("sum Of fourth two elements= %d\n",a[6]/a[7]);
     }
     else {
          printf("Child error!!!\n");
     }
```

}

```
showmik@showmik-virtual-machine:~$ gcc n1n2.c -o n1n2
showmik@showmik-virtual-machine:~$ ./n1n2
23569712sum of first two elements= 5
23569712sum Of third two elements= 63
showmik@showmik-virtual-machine:~$ 23569712sum Of fourth two elements= 0
23569712sum Of second two elements= -1
```