ASSIGNMENT 2 A

NAME:-ANDHALE NITIN PARASRAM

ROLL NO:-I3111

DIV:-1

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

#include<unistd.h>

#include<stdio.h>

void asc_sort(int arr[],int size)

{
int step,i;

        for(step=0; step<size-1;step++)
```

```
        {
                for(i=0;i<size-step-1;i++)
                {
                        if(arr[i]>arr[i+1])
                        {
                                int temp=arr[i];
                                arr[i]=arr[i+1];
                                arr[i+1]=temp;
                        }
                }
        }
}
```

```
void desc_sort(int arr[],int size)



{
int step,i;

        for( step=0; step<size-1;step++)

        {

                for( i=0;i<size-step-1;i++)

                {

                        if(arr[i]<arr[i+1])

                        {

                                int temp=arr[i];

                                arr[i]=arr[i+1];

                                arr[i+1]=temp;

                        }
```

```
                }

        }

}



int main()

{

        int pid,size,arr[size],i;



        printf("enter how much element do you want to be sorted \n");

        scanf("%d",&size);



        printf("enter the elements upto size %d\n",size);

        for(i=0;i<size;i++)
```

```
{
        scanf("%d",&arr[i]);
}



pid=fork();



if(pid==0)
{
      //sleep(5);
        printf("pid of the child is %d\n",getpid());
        printf("pid of the parent is %d\n",getppid());
        asc_sort(arr,size);
        for(i=0;i<size;i++)
```

```
        {

                printf("%d\n",arr[i]);

        }



        //system("ps -el | grep init");









}



else if(pid >0)

{
        system("ps -el|grep Z");

        i=wait(0);
```

```
        printf("Terminated Child's pid is %d\n",i);

        printf("parent's pid is %d\n",getpid());

        printf("parent's parent pid is %d\n",getppid());


        desc_sort(arr,size);

        for(i=0;i<size;i++)

        {

                printf("%d\n",arr[i]);

        }


}


else

{
```

```
                printf("Error in Fork\n");

        }

}
```

```
//OUTPUT FOR ZOMBI STATE
enter how much element do you want to be sorted
4
enter the elements upto size 4
3
2
7
4
7
pid of the child is 18918
pid of the parent is 18917
2
3
4
7
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 Z  1000 18918 18917  0  80   0 -     0 exit   pts/0    00:00:00 assign2 <defunct>
Terminated Child's pid is 18918
```

parent's pid is 18917

parent's parent pid is 7921

7

4

3

2

//OUTPUT FOR ORPHAN STATE

enter how much element do you want to be sorted

4

enter the elements upto size 4

7

2

8

1

3

parent's pid is 13206

parent's parent pid is 7921

8721pid of the child is 13372

pid of the parent is 13206

1

2

7

8

student@student-OptiPlex-390:~/nitin$ 4 S 0 1 0 0 80 0 - 8447 poll_s
? 00:00:01 init

```
4 S  1000  1480  1096  0  80   0 - 10047 poll_s ?        00:00:00 init
```