```c
#include<stdio.h>
#include<sys/wait.h>
#include<unistd.h>
#include<sys/types.h>
#include<stdlib.h>
int main()
{

        pid_t pid;
        pid=fork();
        int a[10]={5,3,4,2,1,9,7,8,6,0};
        int loop;
        if(pid>1)
        {
                printf("This is Parent Process Id is %d\n",(int)getpid());
                printf("Array Sorting begins with parent process in Ascending Order\n");
                for(int i=0;i<10;i++)
                {
                        for(int j=i+1;j<10;j++)
                        {
                                if(a[i]>a[j])
                                {
                                        int temp;
                                        temp=a[i];
                                        a[i]=a[j];
                                        a[j]=temp;
                                }
                        }
                }
                for(loop = 0; loop < 10; loop++)
```

```c
                {
                        printf("%d \n", a[loop]);
                }
                pid=wait(NULL);
                printf("Program Terminated in Parent Process");


        }
else if(pid==0)
{
        printf("This is Child Process Id is %d\n",(int)getpid());
        printf("This is Child Process Parent Id is %d\n",(int)getppid());
        printf("Array Sorting begins with Child process in Descending Order\n");
                for(int i=0;i<10;i++)
                {
                        for(int j=i+1;j<10;j++)
                        {
                                if(a[i]<a[j])
                                {
                                        int temp;
                                        temp=a[i];
                                        a[i]=a[j];
                                        a[j]=temp;
                                }
                        }
                }
                for(loop = 0; loop < 10; loop++)
                {
                        printf("%d \n", a[loop]);
                }
                printf("Child Process Terminated\n");

}
return 0;
}
```