

### **question 1**

```
#include <stdio.h>
```

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

```
#include <unistd.h>
```

```
#include<stdlib.h>
```

```
int main(int argc ,char*argv[]) {
```

```
    int arr[argc-1];
```

```
    int n=argc-1;
```

```
    for(int i=1;i<argc;i++)
```

```
    {
```

```
        arr[i]=atoi(argv[i]);
```

```
    }
```

```
    pid_t pid=fork();
```

```
    if(pid==0)
```

```
    {
```

```
        for(int i=1;i<n;i++)
```

```
        {
```

```
            for(int j=i+1;j<=n;j++)
```

```
            {
```

```
                if(arr[i]>arr[j])
```

```
                {
```

```
                    int a=arr[i];
```

```
                    arr[i]=arr[j];
```

```

        arr[j]=a;
    }
}

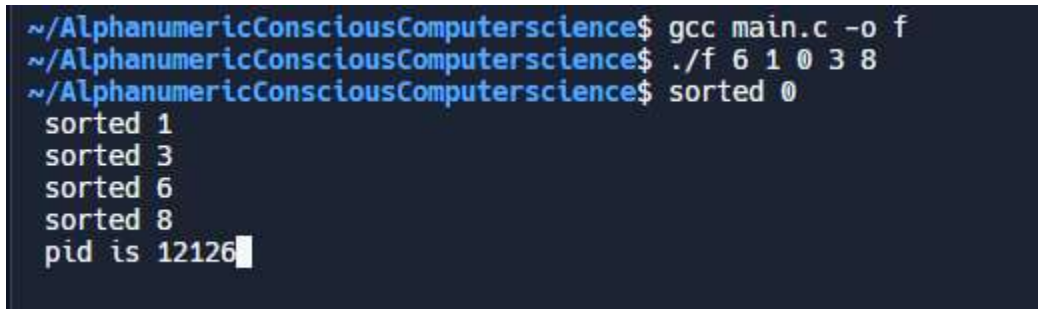
}

for(int i=1;i<=n;i++)
{
    printf("sorted %d\n ",arr[i]);
}

printf("pid is %d",getpid());
}

}

```



```

~/AlphanumericConsciousComputerscience$ gcc main.c -o f
~/AlphanumericConsciousComputerscience$ ./f 6 1 0 3 8
~/AlphanumericConsciousComputerscience$ sorted 0
sorted 1
sorted 3
sorted 6
sorted 8
pid is 12126

```

## **question 2**

```

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <stdlib.h>

int main(int argc ,char*argv[]) {

```

```

int arr[argc-1];

int n=argc-1;

for(int i=1;i<argc;i++)
{
    arr[i]=atoi(argv[i]);
}

pid_t pid=fork();

if(pid==0)
{
    for(int i=1;i<n;i++)
    {
        for(int j=i+1;j<=n;j++)
        {
            if(arr[i]>arr[j])
            {
                int a=arr[i];
                arr[i]=arr[j];
                arr[j]=a;
            }
        }
    }

    for(int i=1;i<=n;i++)
    {
        printf("sorted %d\n ",arr[i]);
    }

    printf("pid is %d",getpid());
}

```

```

    }

    else

    {

        pid_t t= fork();

        if(t==0)

        {

            for(int i = 1; i < n; ++i){

                for(int j = i+1; j <=n ;++j){

                    if(arr[i] < arr[j]){

                        int a = arr[i];

                        arr[i] = arr[j];

                        arr[j] = a;

                    }

                }

            }

            for(int i=1;i<=n;i++)

            {

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

            }

            printf("pid is %d",getpid());

        }

        else{

            printf(" Parent pid is %d",getpid());

        }

    }

}

}

```

```
~/AlphanumericConsciousComputerscience$ ./gg 6 7 0 2 4 9
Parent pid is 19903sorted 9
sorted 7
sorted 6
sorted 4
sorted 2
sorted 0
pid is 19905sorted 0
sorted 2
sorted 4
sorted 6
sorted 7
sorted 9
pid is 19904~/AlphanumericConsciousComputerscience$
```

### Question 3

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

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

```
#include <unistd.h>
```

```
int main(int argc, char *argv[]) {
```

```
    int arr[argc - 1];
```

```
    int n = argc - 1;
```

```
    for (int i = 1; i < argc; i++) {
```

```
        arr[i] = atoi(argv[i]);
```

```
    }
```

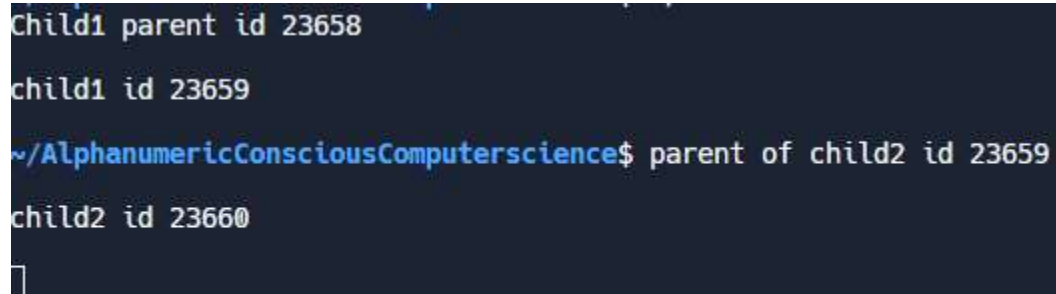
```
    pid_t pid = fork();
```

```
    if (pid == 0) {
```

```
        printf("child1 id %d", getpid());
```

```
        printf("\n\n");
```

```
pid_t i = fork();  
if (i == 0) {  
    printf("child2 id %d", getpid());  
    printf("\n\n");  
} else {  
    printf("parent of child2 id %d", getpid());  
    printf("\n\n");  
}  
} else {  
    printf("Child1 parent id %d", getpid());  
    printf("\n\n");  
}  
}
```

A terminal window with a dark background and light blue text. The output of the program is displayed line by line. The first line is "Child1 parent id 23658". The second line is "child1 id 23659". The third line is a prompt "~ /AlphanumericConsciousComputerscience\$ " followed by "parent of child2 id 23659". The fourth line is "child2 id 23660". The prompt character is a small square.

```
Child1 parent id 23658  
child1 id 23659  
~/AlphanumericConsciousComputerscience$ parent of child2 id 23659  
child2 id 23660  
□
```