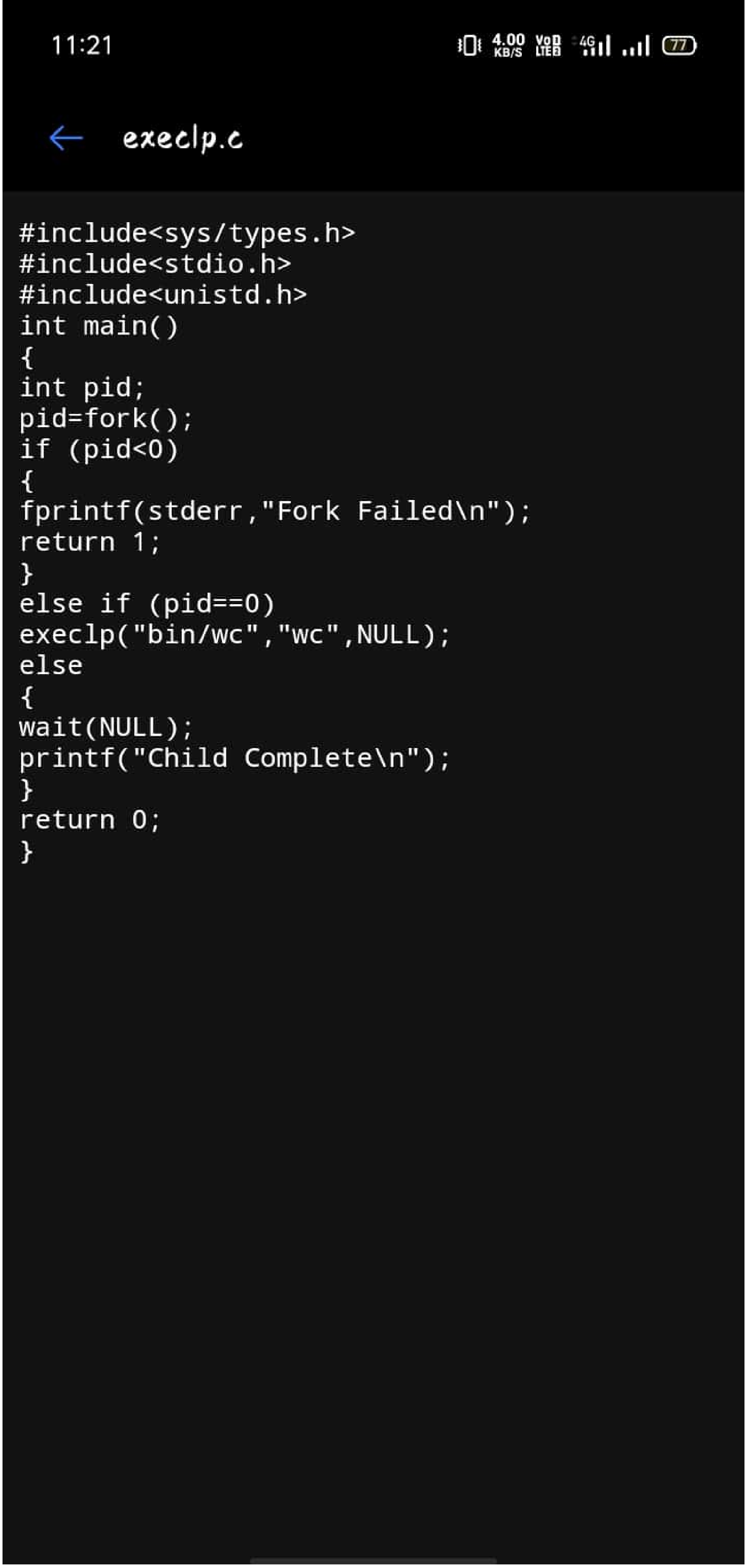
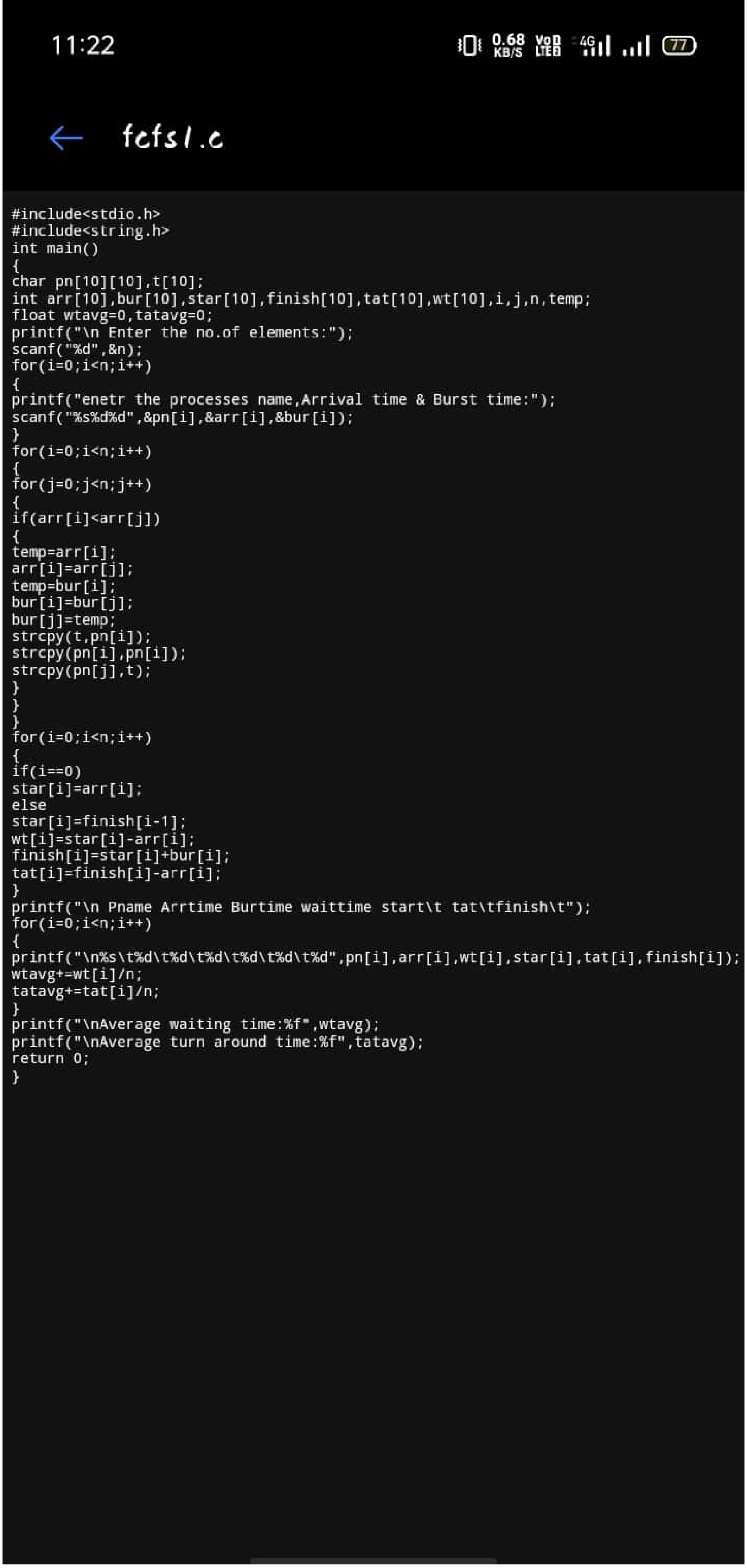


11:21

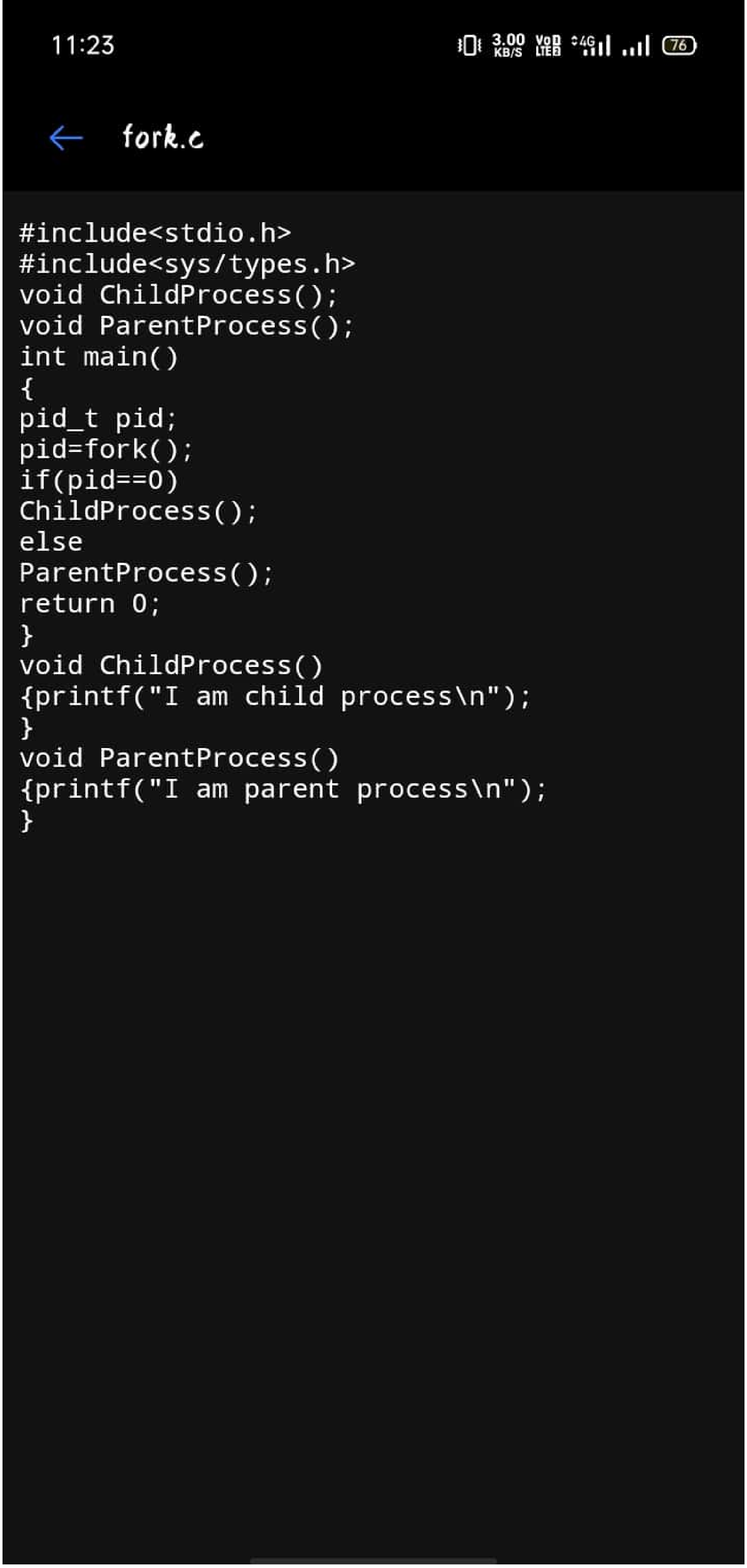
execlp.c

```
#include<sys/types.h>
#include<stdio.h>
#include<unistd.h>
int main()
{
int pid;
pid=fork();
if (pid<0)
{
fprintf(stderr,"Fork Failed\n");
return 1;
}
else if (pid==0)
execlp("bin/wc","wc",NULL);
else
{
wait(NULL);
printf("Child Complete\n");
}
return 0;
}
```

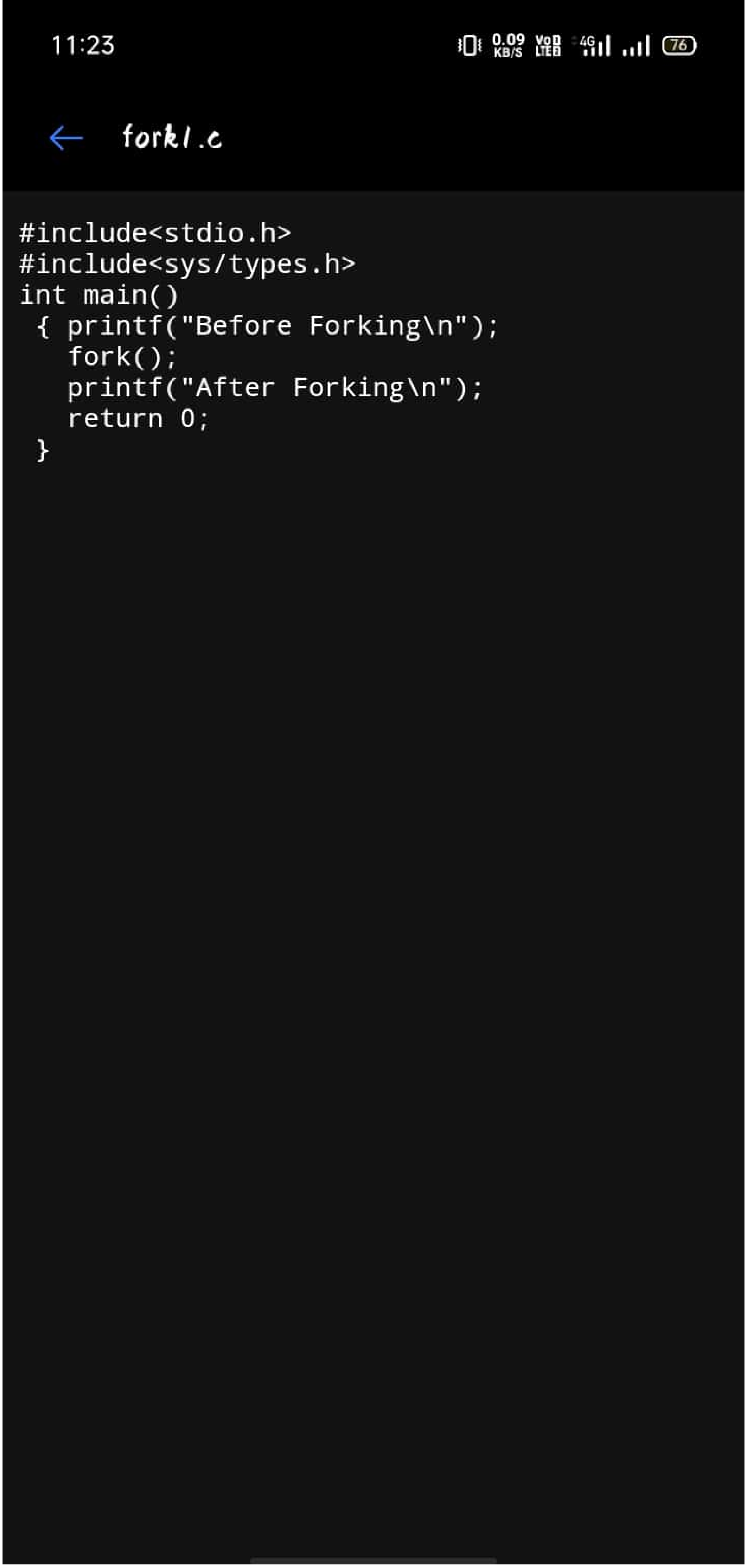


```
#include<stdio.h>
#include<string.h>
int main()
{
char pn[10][10],t[10];
int arr[10],bur[10],star[10],finish[10],tat[10],wt[10],i,j,n,temp;
float wtavg=0,tatavg=0;
printf("\n Enter the no.of elements:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("enetr the processes name,Arrival time & Burst time:");
scanf("%s%d%d",&pn[i],&arr[i],&bur[i]);
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(arr[i]<arr[j])
{
temp=arr[i];
arr[i]=arr[j];
temp=bur[i];
bur[i]=bur[j];
bur[j]=temp;
strcpy(t,pn[i]);
strcpy(pn[i],pn[i]);
strcpy(pn[j],t);
}
}
}
for(i=0;i<n;i++)
{
if(i==0)
star[i]=arr[i];
else
star[i]=finish[i-1];
wt[i]=star[i]-arr[i];
finish[i]=star[i]+bur[i];
tat[i]=finish[i]-arr[i];
}
printf("\n Pname Arrtime Burtime waittime start\t tat\tfinish\t");
for(i=0;i<n;i++)
{
printf("\n%s\t%d\t%d\t%d\t%d\t%d\t%d",pn[i],arr[i],wt[i],star[i],tat[i],finish[i]);
wtavg+=wt[i]/n;
tatavg+=tat[i]/n;
}
printf("\nAverage waiting time:%f",wtavg);
printf("\nAverage turn around time:%f",tatavg);
return 0;
}
```

Scanned with CamScanner



```c
#include<stdio.h>
#include<sys/types.h>
void ChildProcess();
void ParentProcess();
int main()
{
pid_t pid;
pid=fork();
if(pid==0)
ChildProcess();
else
ParentProcess();
return 0;
}
void ChildProcess()
{printf("I am child process\n");
}
void ParentProcess()
{printf("I am parent process\n");
}
```

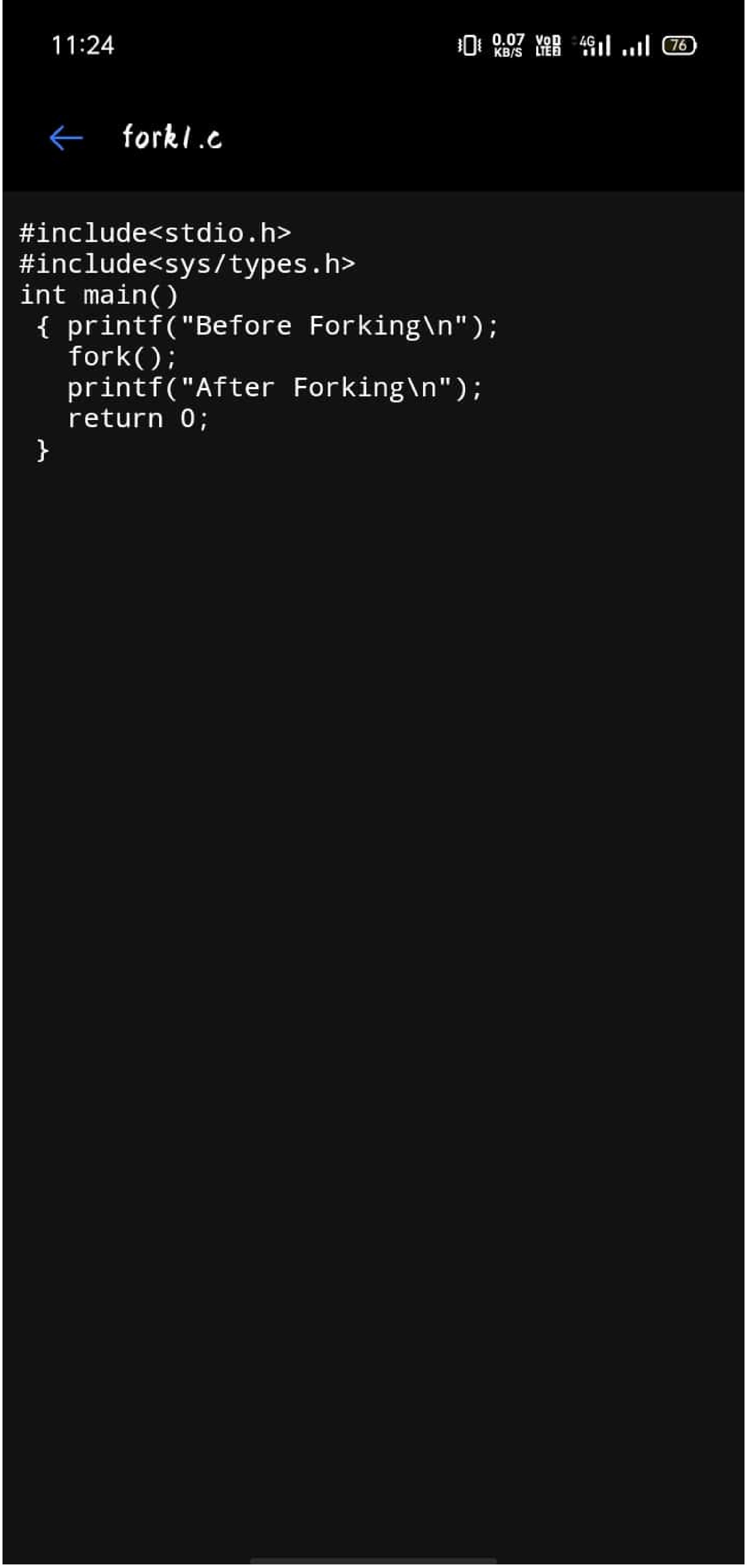


11:23

0.09 KB/S VoB LTE 4G 76

fork1.c

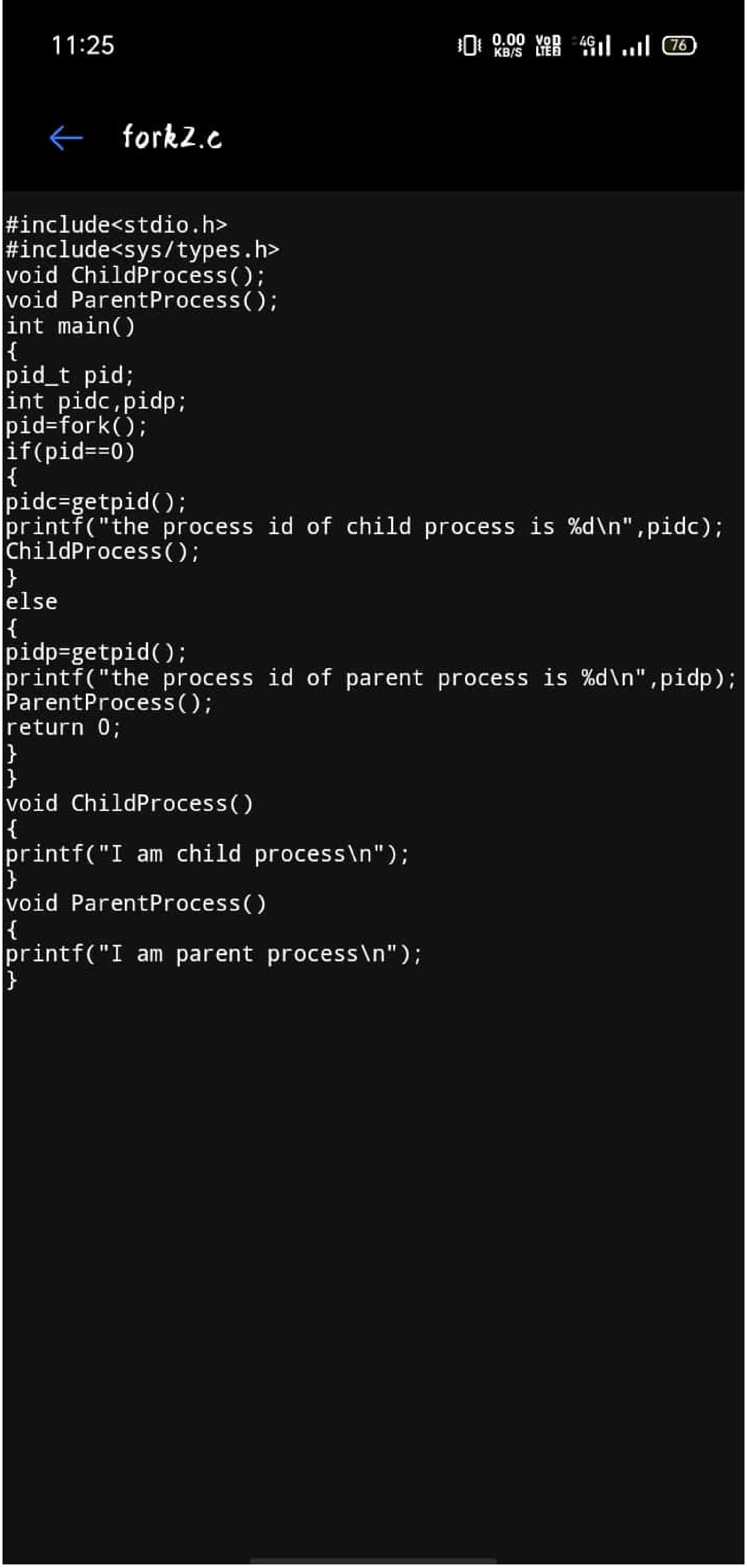
```c
#include<stdio.h>
#include<sys/types.h>
int main()
 { printf("Before Forking\n");
   fork();
   printf("After Forking\n");
   return 0;
 }
```



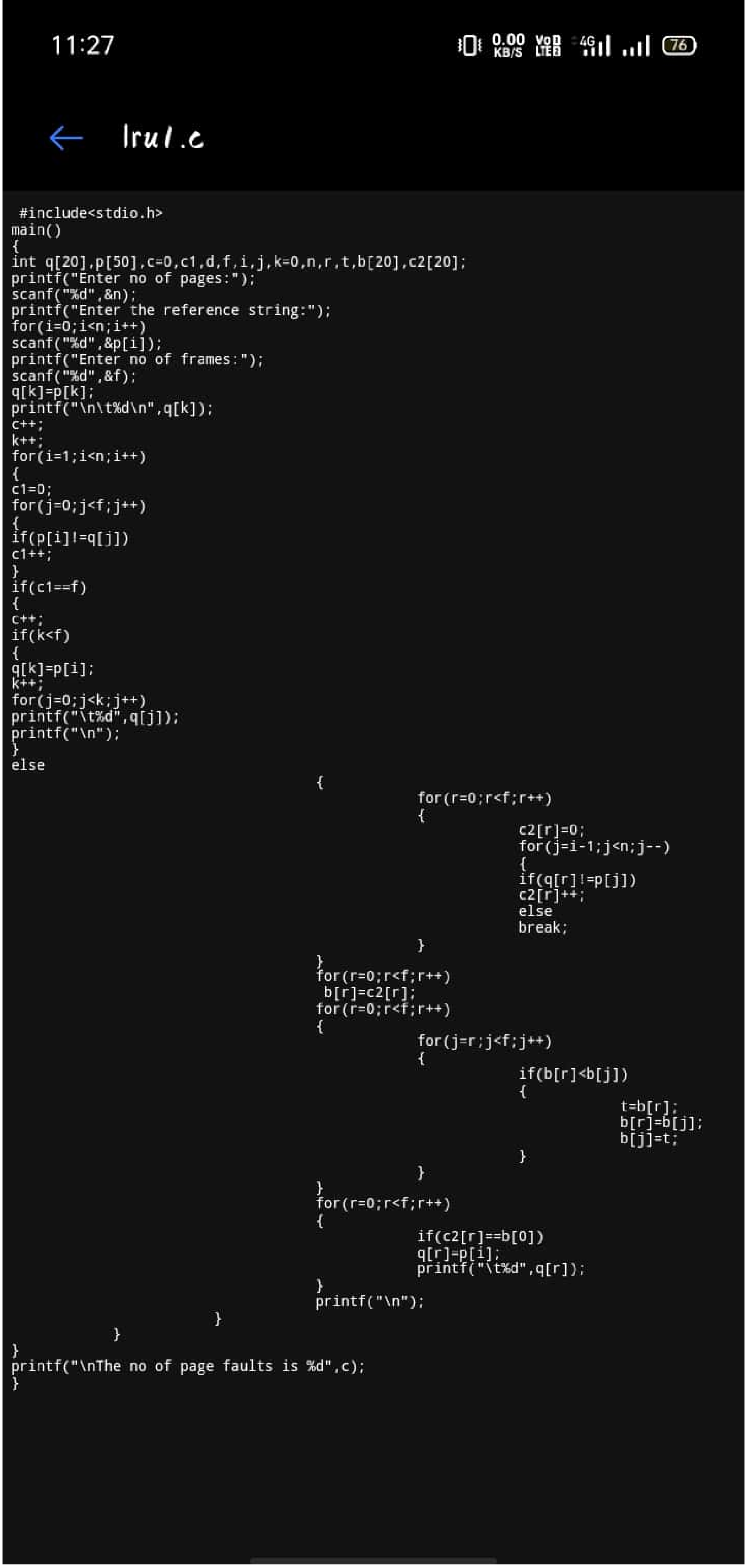
11:24

fork1.c

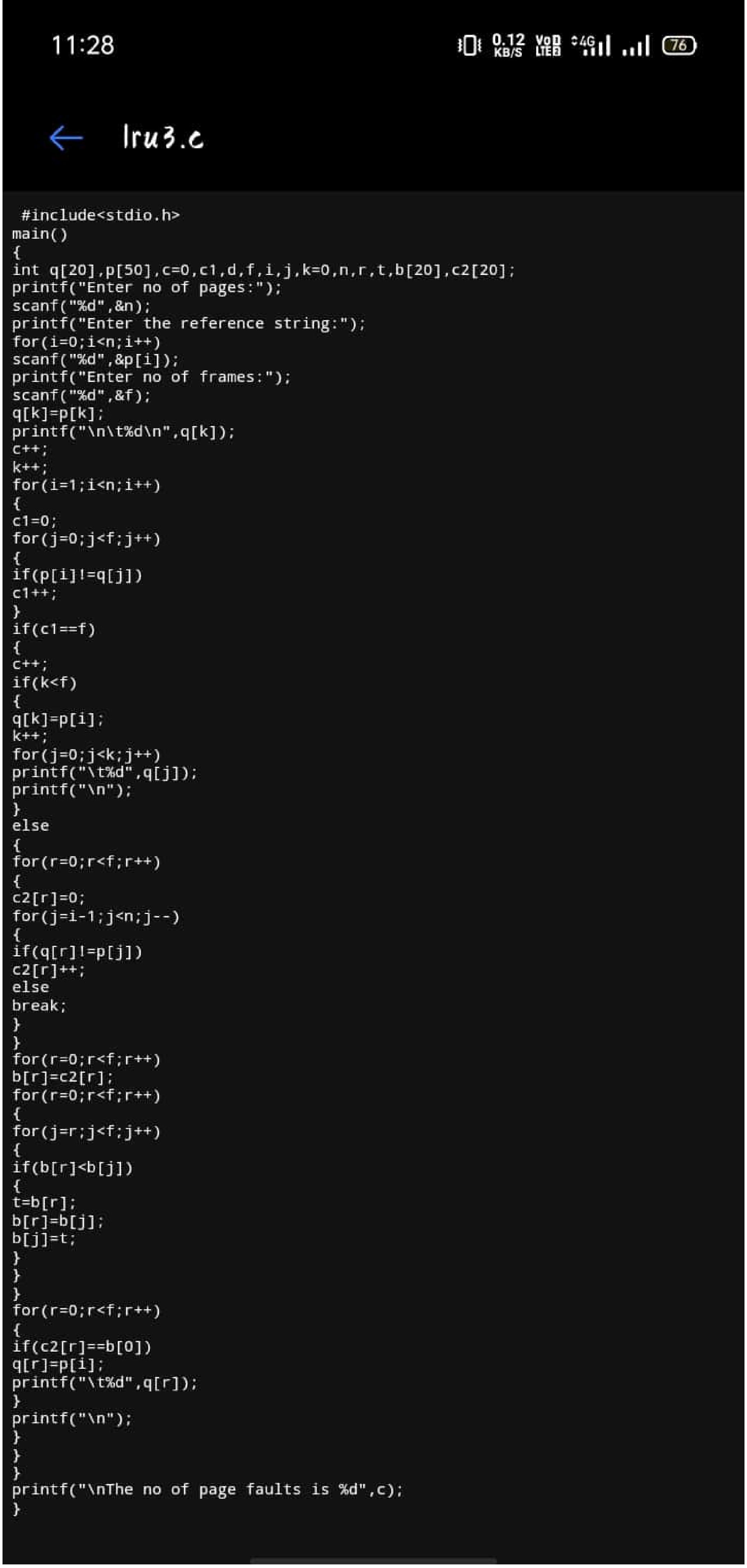
```
#include<stdio.h>
#include<sys/types.h>
int main()
 { printf("Before Forking\n");
   fork();
   printf("After Forking\n");
   return 0;
 }
```



```c
#include<stdio.h>
#include<sys/types.h>
void ChildProcess();
void ParentProcess();
int main()
{
pid_t pid;
int pidc,pidp;
pid=fork();
if(pid==0)
{
pidc=getpid();
printf("the process id of child process is %d\n",pidc);
ChildProcess();
}
else
{
pidp=getpid();
printf("the process id of parent process is %d\n",pidp);
ParentProcess();
return 0;
}
}
void ChildProcess()
{
printf("I am child process\n");
}
void ParentProcess()
{
printf("I am parent process\n");
}
```

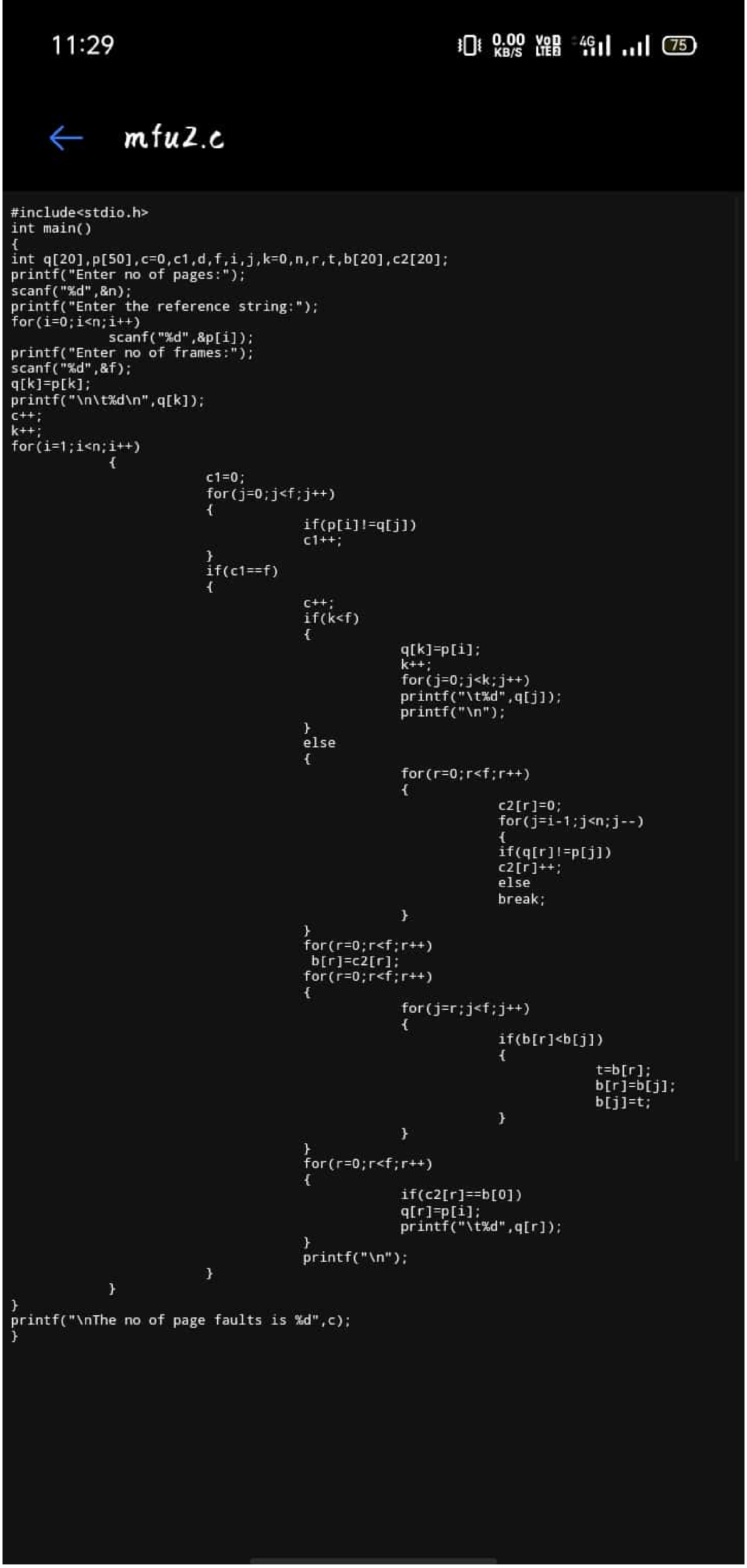


```
 #include<stdio.h>
main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
scanf("%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
{
c1=0;
for(j=0;j<f;j++)
{
if(p[i]!=q[j])
c1++;
}
if(c1==f)
{
c++;
if(k<f)
{
q[k]=p[i];
k++;
for(j=0;j<k;j++)
printf("\t%d",q[j]);
printf("\n");
}
else
                                    {
                                                for(r=0;r<f;r++)
                                                {
                                                            c2[r]=0;
                                                            for(j=i-1;j<n;j--)
                                                            {
                                                            if(q[r]!=p[j])
                                                            c2[r]++;
                                                            else
                                                            break;
                                                }
                                    }
                                    for(r=0;r<f;r++)
                                     b[r]=c2[r];
                                    for(r=0;r<f;r++)
                                    {
                                                for(j=r;j<f;j++)
                                                {
                                                            if(b[r]<b[j])
                                                            {
                                                                        t=b[r];
                                                                        b[r]=b[j];
                                                                        b[j]=t;
                                                            }
                                                }
                                    }
                                    for(r=0;r<f;r++)
                                    {
                                                if(c2[r]==b[0])
                                                q[r]=p[i];
                                                printf("\t%d",q[r]);
                                    }
                                    printf("\n");
                        }
            }
}
printf("\nThe no of page faults is %d",c);
}
```


11:28

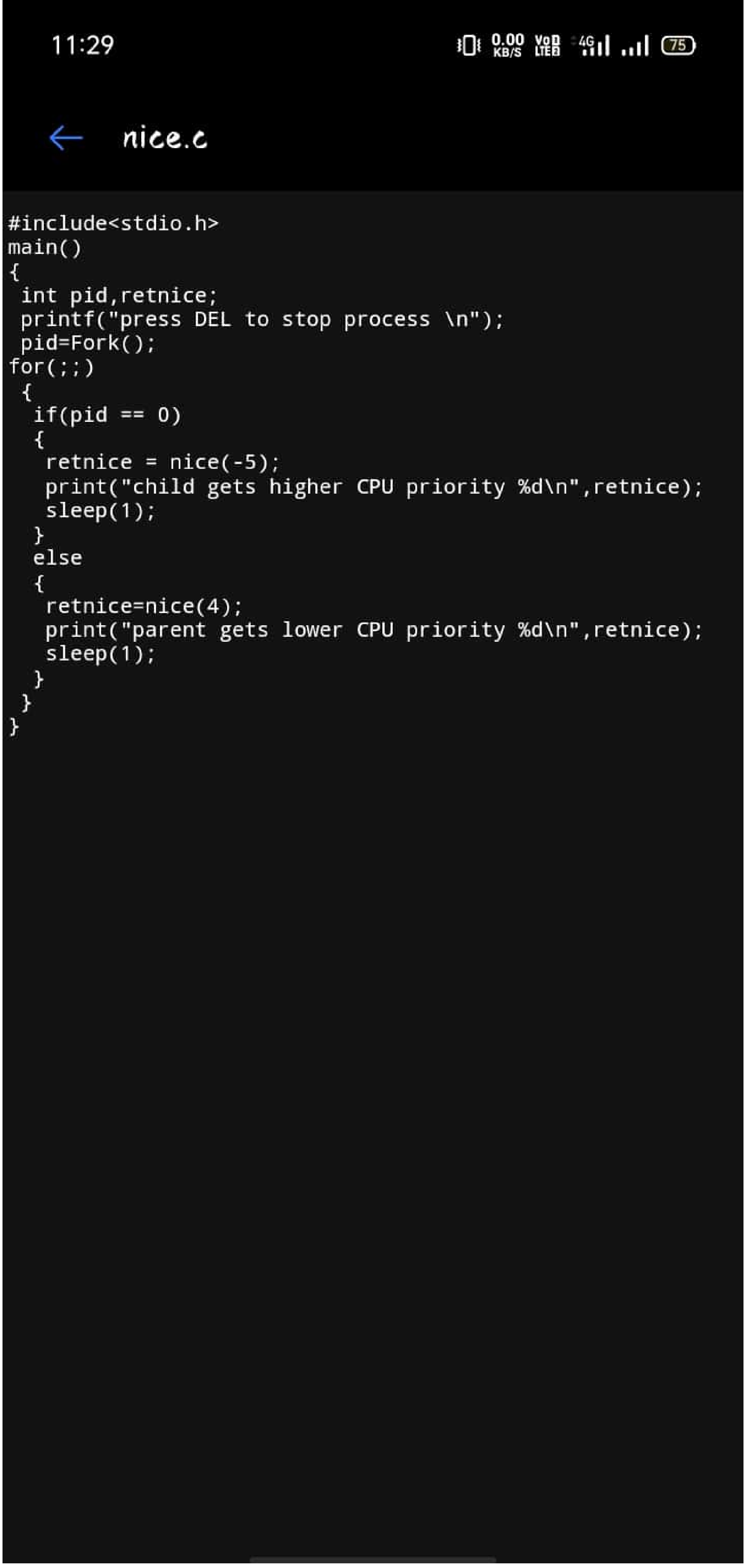
lru3.c

```
 #include<stdio.h>
main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
scanf("%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
{
c1=0;
for(j=0;j<f;j++)
{
if(p[i]!=q[j])
c1++;
}
if(c1==f)
{
c++;
if(k<f)
{
q[k]=p[i];
k++;
for(j=0;j<k;j++)
printf("\t%d",q[j]);
printf("\n");
}
else
{
for(r=0;r<f;r++)
{
c2[r]=0;
for(j=i-1;j<n;j--)
{
if(q[r]!=p[j])
c2[r]++;
else
break;
}
}
for(r=0;r<f;r++)
b[r]=c2[r];
for(r=0;r<f;r++)
{
for(j=r;j<f;j++)
{
if(b[r]<b[j])
{
t=b[r];
b[r]=b[j];
b[j]=t;
}
}
}
for(r=0;r<f;r++)
{
if(c2[r]==b[0])
q[r]=p[i];
printf("\t%d",q[r]);
}
printf("\n");
}
}
}
printf("\nThe no of page faults is %d",c);
}
```


11:29

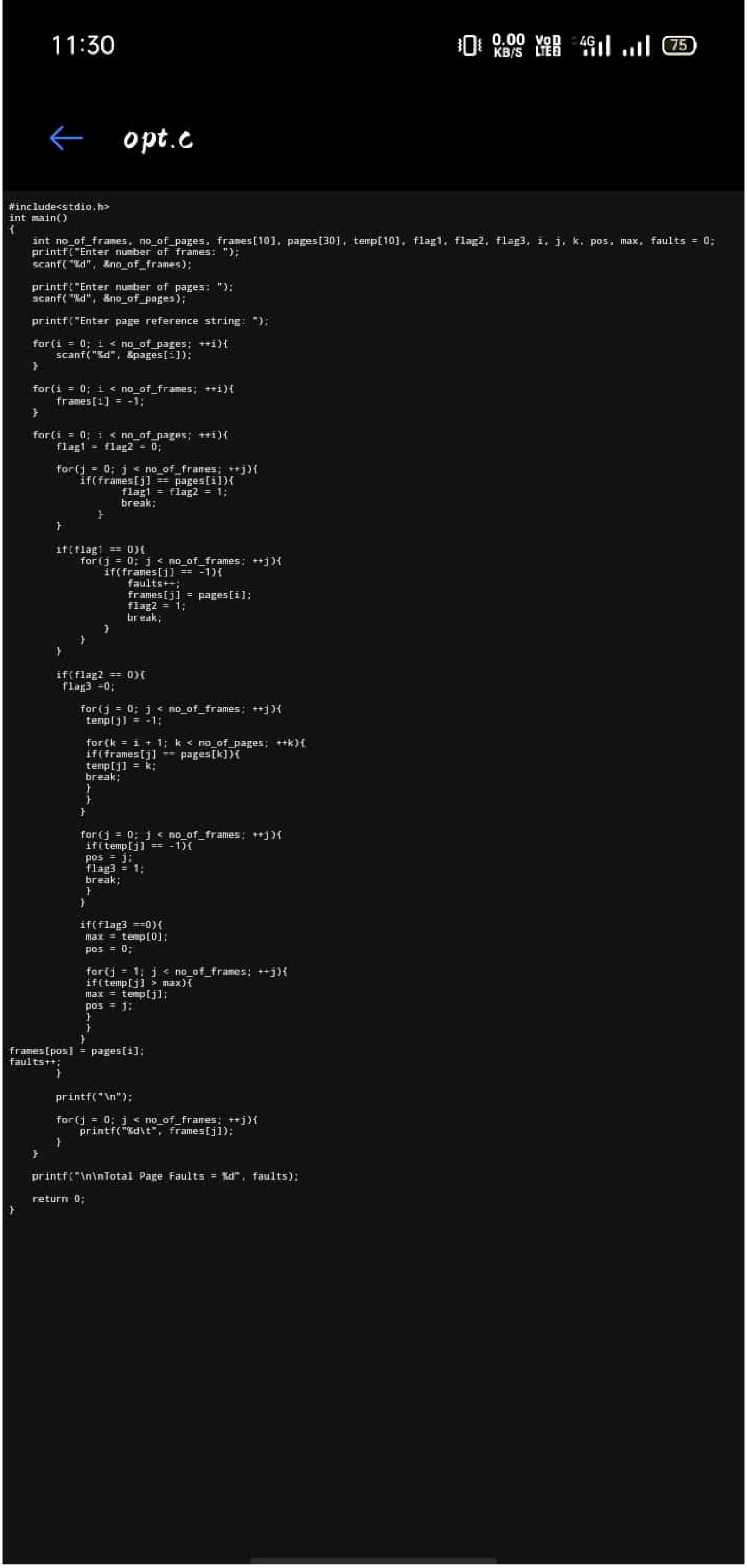
mfu2.c

```
#include<stdio.h>
int main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
        scanf("%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
        {
                c1=0;
                for(j=0;j<f;j++)
                {
                        if(p[i]!=q[j])
                        c1++;
                }
                if(c1==f)
                {
                        c++;
                        if(k<f)
                        {
                                q[k]=p[i];
                                k++;
                                for(j=0;j<k;j++)
                                printf("\t%d",q[j]);
                                printf("\n");
                        }
                        else
                        {
                                for(r=0;r<f;r++)
                                {
                                        c2[r]=0;
                                        for(j=i-1;j<n;j--)
                                        {
                                        if(q[r]!=p[j])
                                        c2[r]++;
                                        else
                                        break;
                                }
                        }
                        for(r=0;r<f;r++)
                         b[r]=c2[r];
                        for(r=0;r<f;r++)
                        {
                                for(j=r;j<f;j++)
                                {
                                        if(b[r]<b[j])
                                        {
                                                t=b[r];
                                                b[r]=b[j];
                                                b[j]=t;
                                        }
                                }
                        }
                        for(r=0;r<f;r++)
                        {
                                if(c2[r]==b[0])
                                q[r]=p[i];
                                printf("\t%d",q[r]);
                        }
                        printf("\n");
                }
        }
}
printf("\nThe no of page faults is %d",c);
}
```

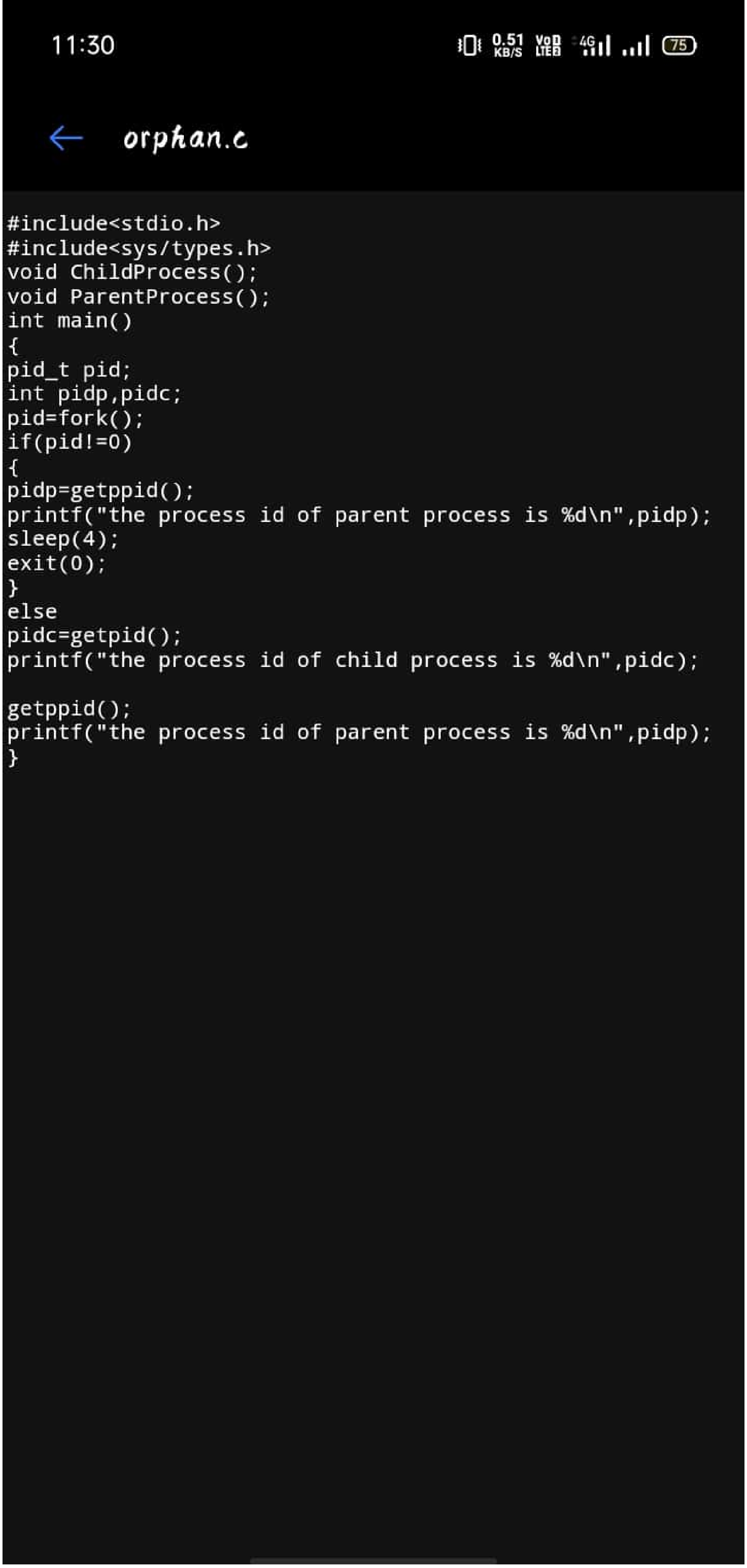

11:29

nice.c

```
#include<stdio.h>
main()
{
 int pid,retnice;
 printf("press DEL to stop process \n");
 pid=Fork();
for(;;)
 {
  if(pid == 0)
  {
   retnice = nice(-5);
   print("child gets higher CPU priority %d\n",retnice);
   sleep(1);
  }
  else
  {
   retnice=nice(4);
   print("parent gets lower CPU priority %d\n",retnice);
   sleep(1);
  }
 }
}
```

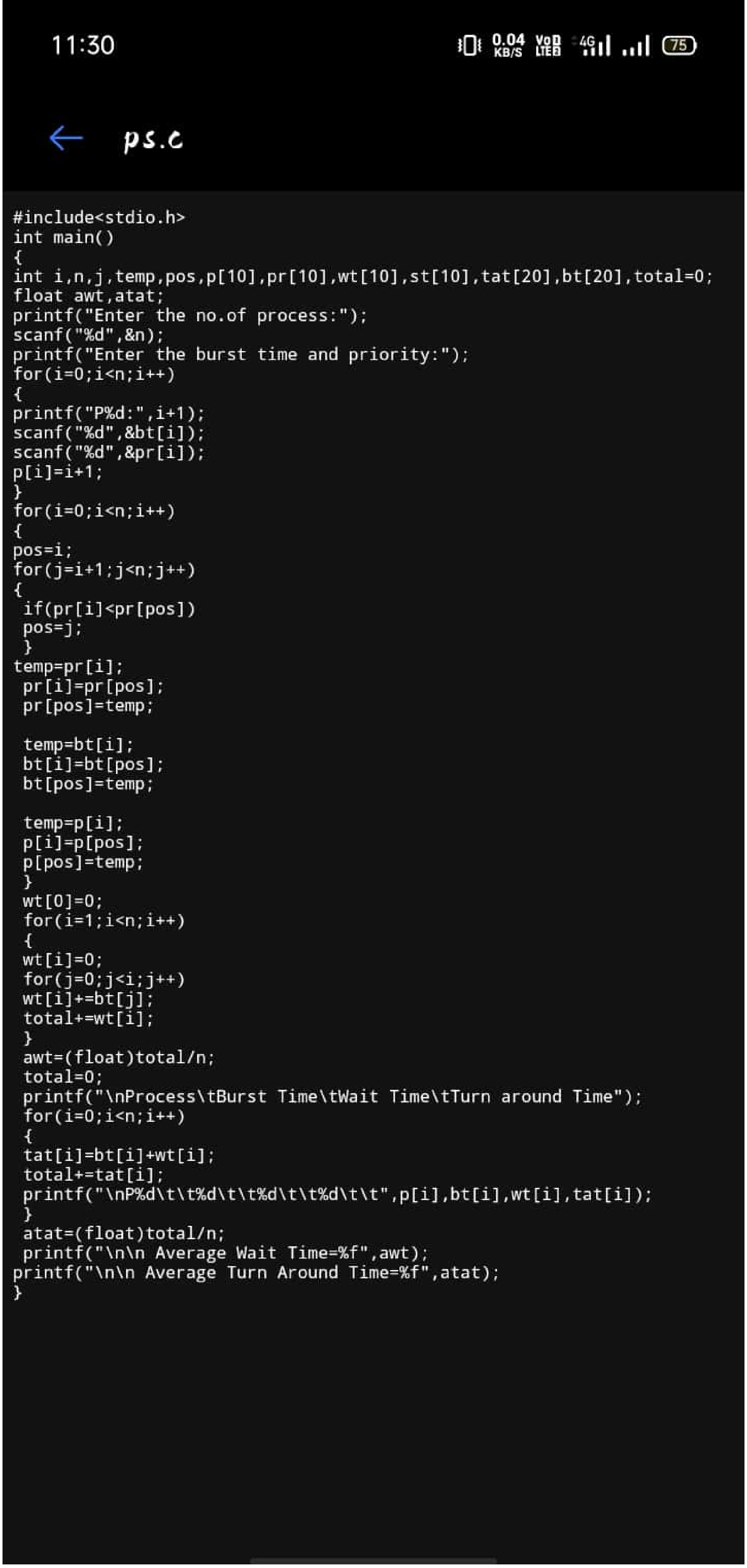


```c
#include<stdio.h>
int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos, max, faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);

    printf("Enter page reference string: ");

    for(i = 0; i < no_of_pages; ++i){
        scanf("%d", &pages[i]);
    }

    for(i = 0; i < no_of_frames; ++i){
        frames[i] = -1;
    }

    for(i = 0; i < no_of_pages; ++i){
        flag1 = flag2 = 0;

        for(j = 0; j < no_of_frames; ++j){
            if(frames[j] == pages[i]){
                   flag1 = flag2 = 1;
                   break;
               }
        }

        if(flag1 == 0){
            for(j = 0; j < no_of_frames; ++j){
                if(frames[j] == -1){
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }

        if(flag2 == 0){
         flag3 =0;

            for(j = 0; j < no_of_frames; ++j){
             temp[j] = -1;

             for(k = i + 1; k < no_of_pages; ++k){
             if(frames[j] == pages[k]){
             temp[j] = k;
             break;
             }
             }
            }

            for(j = 0; j < no_of_frames; ++j){
             if(temp[j] == -1){
             pos = j;
             flag3 = 1;
             break;
             }
            }

            if(flag3 ==0){
             max = temp[0];
             pos = 0;

             for(j = 1; j < no_of_frames; ++j){
             if(temp[j] > max){
             max = temp[j];
             pos = j;
             }
             }
            }
frames[pos] = pages[i];
faults++;
        }

        printf("\n");

        for(j = 0; j < no_of_frames; ++j){
            printf("%d\t", frames[j]);
        }
    }

    printf("\n\nTotal Page Faults = %d", faults);

    return 0;
}
```


11:30

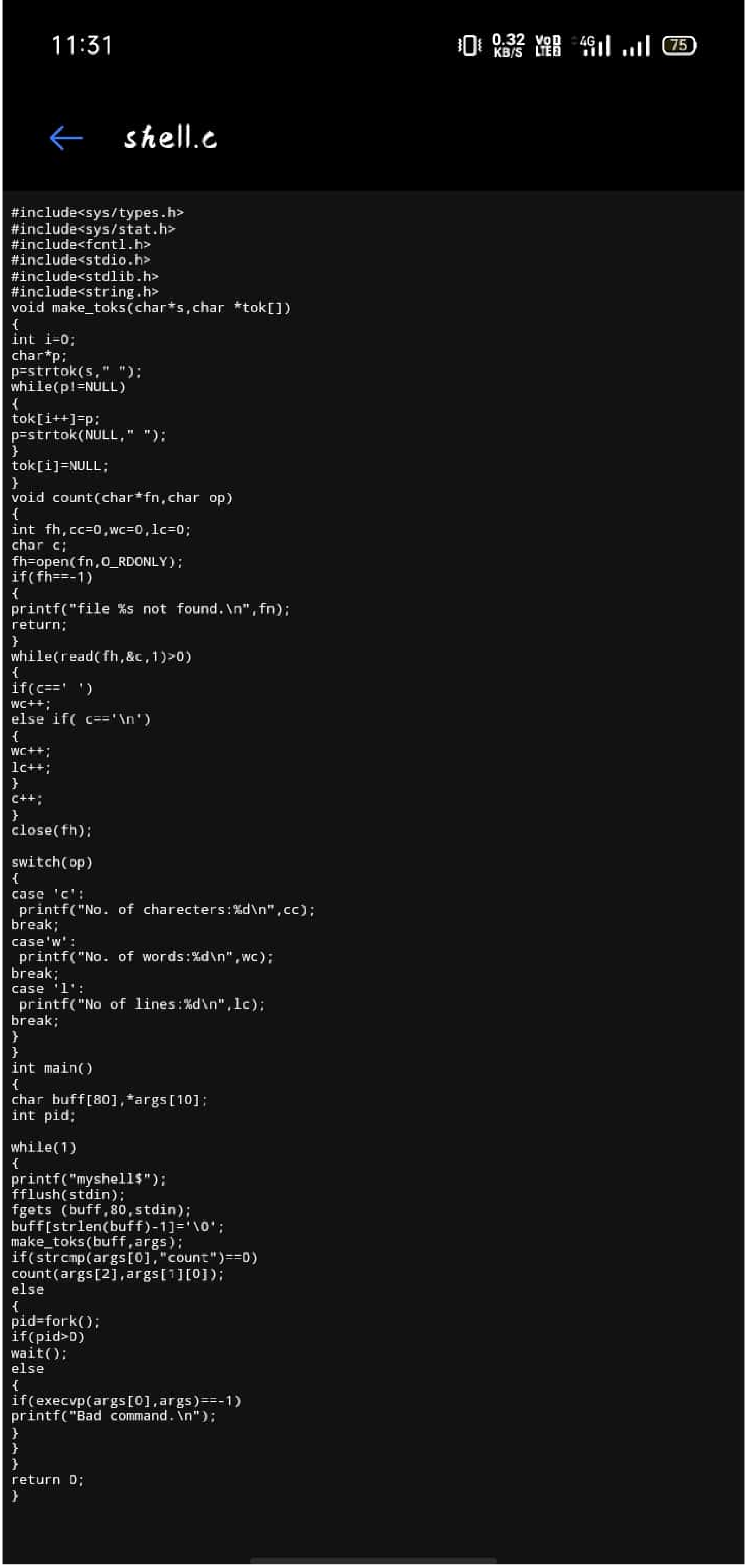
orphan.c

```
#include<stdio.h>
#include<sys/types.h>
void ChildProcess();
void ParentProcess();
int main()
{
pid_t pid;
int pidp,pidc;
pid=fork();
if(pid!=0)
{
pidp=getppid();
printf("the process id of parent process is %d\n",pidp);
sleep(4);
exit(0);
}
else
pidc=getpid();
printf("the process id of child process is %d\n",pidc);

getppid();
printf("the process id of parent process is %d\n",pidp);
}
```


11:30

ps.c

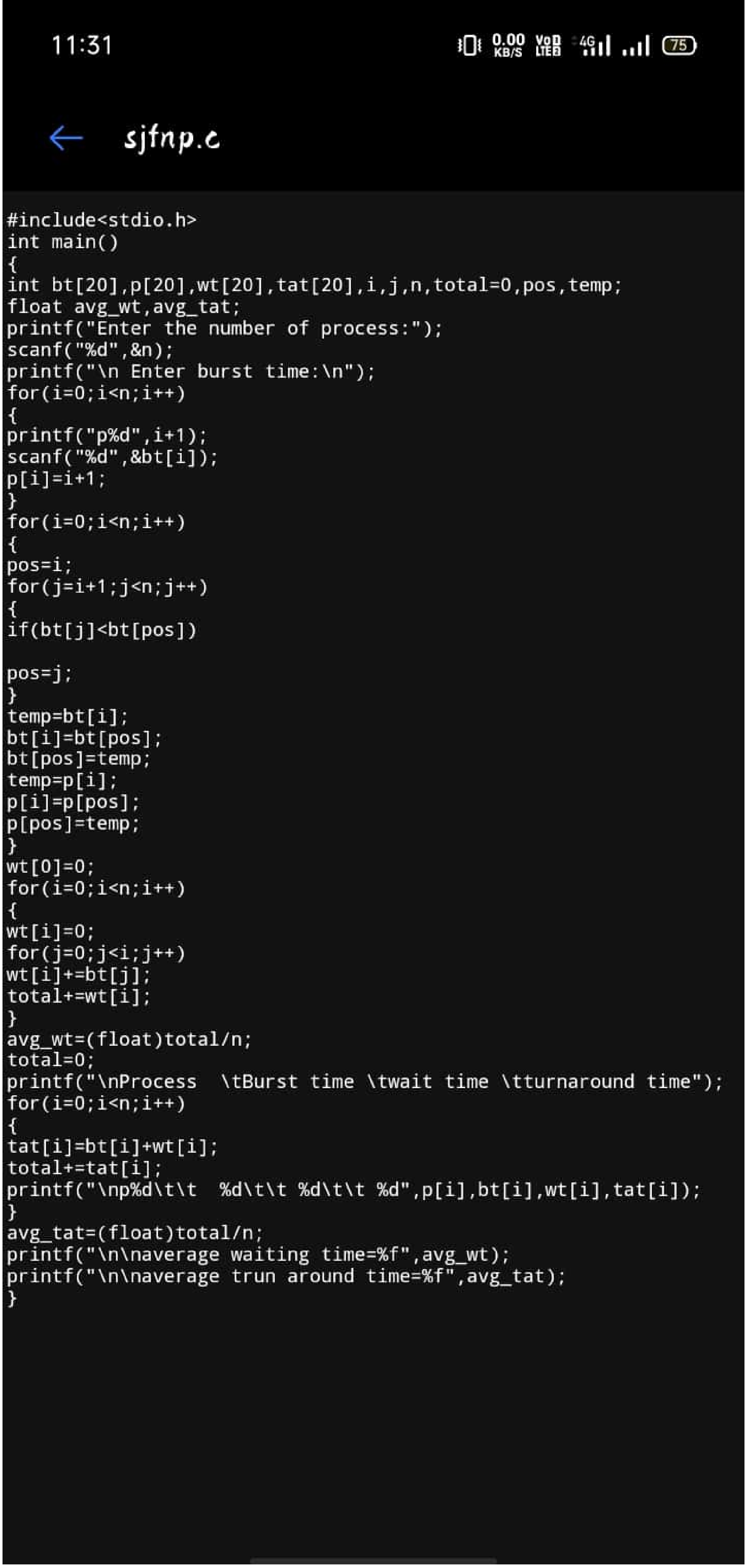
```
#include<stdio.h>
int main()
{
int i,n,j,temp,pos,p[10],pr[10],wt[10],st[10],tat[20],bt[20],total=0;
float awt,atat;
printf("Enter the no.of process:");
scanf("%d",&n);
printf("Enter the burst time and priority:");
for(i=0;i<n;i++)
{
printf("P%d:",i+1);
scanf("%d",&bt[i]);
scanf("%d",&pr[i]);
p[i]=i+1;
}
for(i=0;i<n;i++)
{
pos=i;
for(j=i+1;j<n;j++)
{
 if(pr[i]<pr[pos])
 pos=j;
 }
temp=pr[i];
 pr[i]=pr[pos];
 pr[pos]=temp;

 temp=bt[i];
 bt[i]=bt[pos];
 bt[pos]=temp;

 temp=p[i];
 p[i]=p[pos];
 p[pos]=temp;
 }
 wt[0]=0;
 for(i=1;i<n;i++)
 {
 wt[i]=0;
 for(j=0;j<i;j++)
 wt[i]+=bt[j];
 total+=wt[i];
 }
 awt=(float)total/n;
 total=0;
 printf("\nProcess\tBurst Time\tWait Time\tTurn around Time");
 for(i=0;i<n;i++)
 {
 tat[i]=bt[i]+wt[i];
 total+=tat[i];
 printf("\nP%d\t\t%d\t\t%d\t\t%d\t\t",p[i],bt[i],wt[i],tat[i]);
 }
 atat=(float)total/n;
 printf("\n\n Average Wait Time=%f",awt);
printf("\n\n Average Turn Around Time=%f",atat);
}
```



11:31

shell.c

```
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void make_toks(char*s,char *tok[])
{
int i=0;
char*p;
p=strtok(s," ");
while(p!=NULL)
{
tok[i++]=p;
p=strtok(NULL," ");
}
tok[i]=NULL;
}
void count(char*fn,char op)
{
int fh,cc=0,wc=0,lc=0;
char c;
fh=open(fn,O_RDONLY);
if(fh==-1)
{
printf("file %s not found.\n",fn);
return;
}
while(read(fh,&c,1)>0)
{
if(c==' ')
wc++;
else if( c=='\n')
{
wc++;
lc++;
}
c++;
}
close(fh);

switch(op)
{
case 'c':
 printf("No. of charecters:%d\n",cc);
break;
case'w':
 printf("No. of words:%d\n",wc);
break;
case 'l':
 printf("No of lines:%d\n",lc);
break;
}
}
int main()
{
char buff[80],*args[10];
int pid;

while(1)
{
printf("myshell$");
fflush(stdin);
fgets (buff,80,stdin);
buff[strlen(buff)-1]='\0';
make_toks(buff,args);
if(strcmp(args[0],"count")==0)
count(args[2],args[1][0]);
else
{
pid=fork();
if(pid>0)
wait();
else
{
if(execvp(args[0],args)==-1)
printf("Bad command.\n");
}
}
}
return 0;
}
```



```
#include<stdio.h>
int main()
{
int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
float avg_wt,avg_tat;
printf("Enter the number of process:");
scanf("%d",&n);
printf("\n Enter burst time:\n");
for(i=0;i<n;i++)
{
printf("p%d",i+1);
scanf("%d",&bt[i]);
p[i]=i+1;
}
for(i=0;i<n;i++)
{
pos=i;
for(j=i+1;j<n;j++)
{
if(bt[j]<bt[pos])

pos=j;
}
temp=bt[i];
bt[i]=bt[pos];
bt[pos]=temp;
temp=p[i];
p[i]=p[pos];
p[pos]=temp;
}
wt[0]=0;
for(i=0;i<n;i++)
{
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
total+=wt[i];
}
avg_wt=(float)total/n;
total=0;
printf("\nProcess  \tBurst time \twait time \tturnaround time");
for(i=0;i<n;i++)
{
tat[i]=bt[i]+wt[i];
total+=tat[i];
printf("\np%d\t\t  %d\t\t %d\t\t %d",p[i],bt[i],wt[i],tat[i]);
}
avg_tat=(float)total/n;
printf("\n\naverage waiting time=%f",avg_wt);
printf("\n\naverage trun around time=%f",avg_tat);
}
```