

FA20-BSE-086

UMAIR ASAD

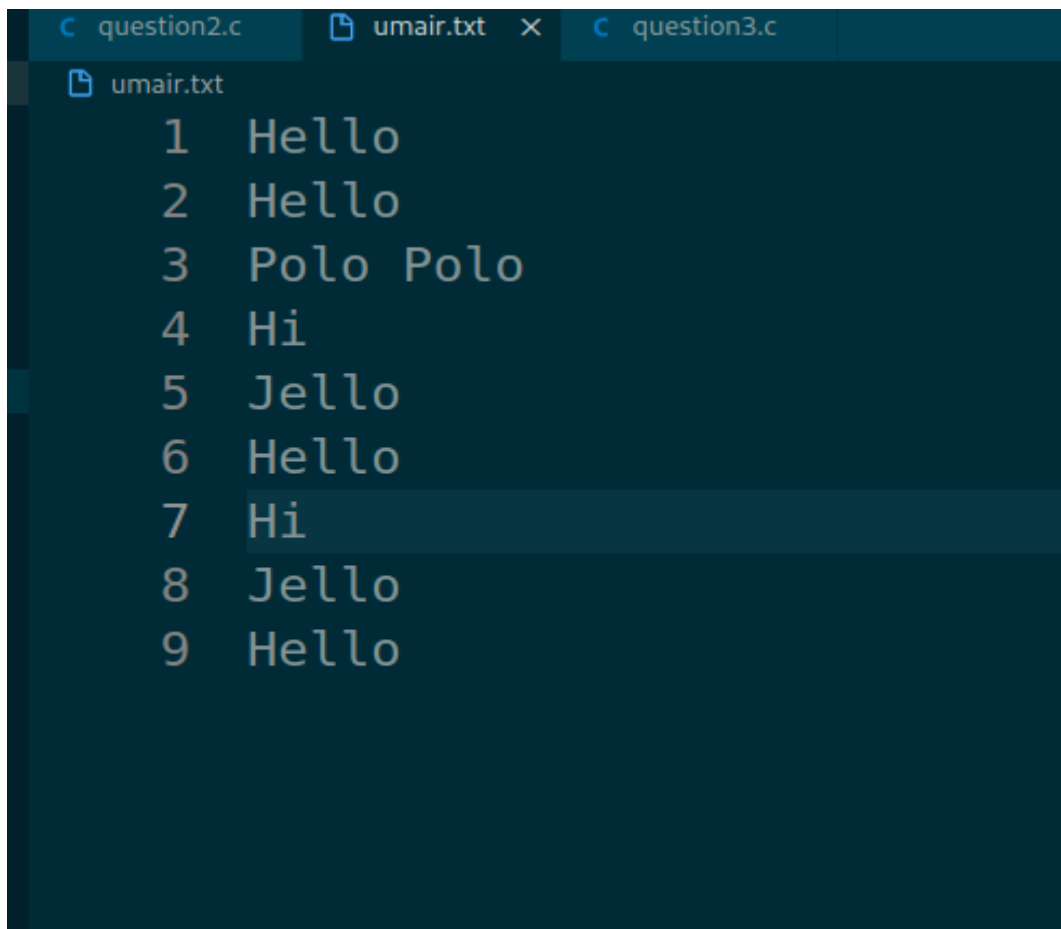
MID TERM EXAM

OS LAB

SEC A

Question 1:

Umain.txt

A screenshot of a code editor window with three tabs: 'question2.c', 'umair.txt', and 'question3.c'. The 'umair.txt' tab is active, showing a list of nine lines of text. The text is as follows:

```
1 Hello
2 Hello
3 Polo Polo
4 Hi
5 Jello
6 Hello
7 Hi
8 Jello
9 Hello
```

The line '7 Hi' is highlighted with a light blue background.

Question 2:

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

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

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

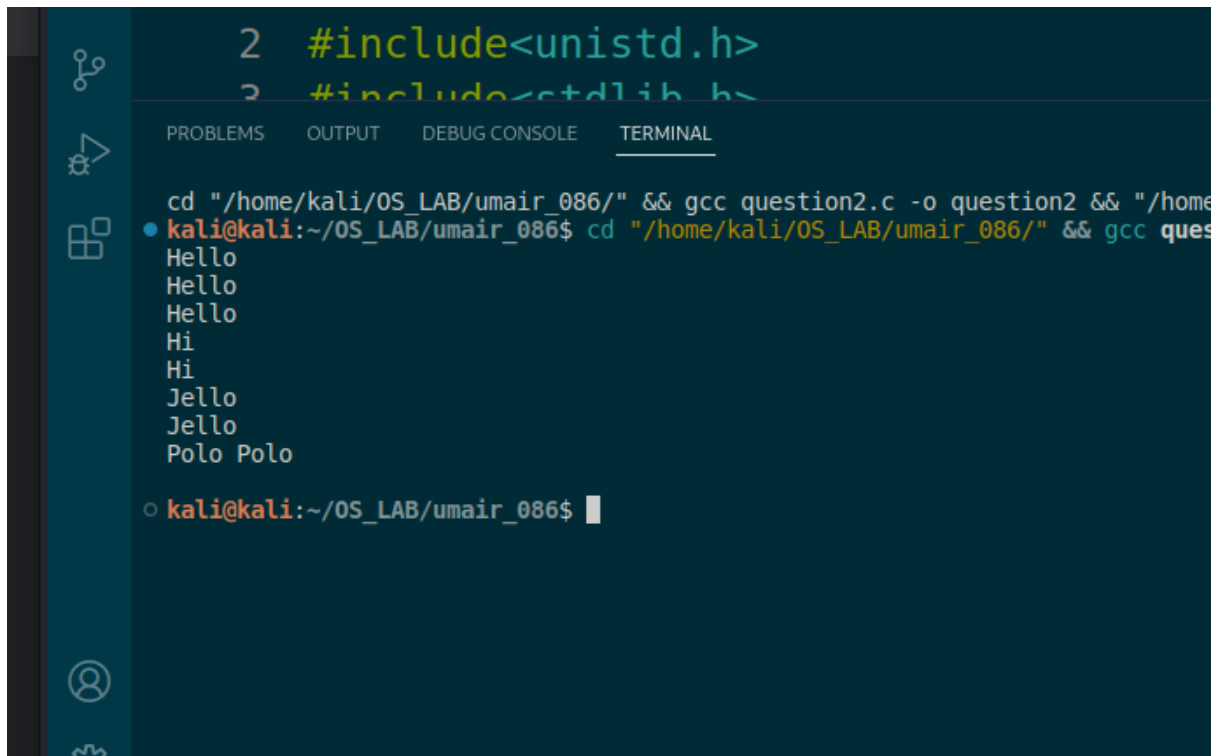
```
#include<sys/types.h>
```

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

//command to use
//uniq -w 2 umair.txt | sort
int main(int argc,char* argv[])
{
    system("uniq -w 2 umair.txt > temp.txt"); // USING 1ST COMMAND
    int pid=fork();
    if(pid!=0){
        wait(NULL);
    }
    if(pid==0)
    {
        execl("/bin/sort","/bin/sort","temp.txt",NULL);//USING SECOND COMMAND
        //cannot execute other command

    }
    return 0;
}
```

Output:



The screenshot shows a code editor with a dark theme. The top part displays C code with two lines: `2 #include<unistd.h>` and `3 #include<stdlib.h>`. Below the code, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active, showing the command `cd "/home/kali/OS_LAB/umair_086/" && gcc question2.c -o question2 && "/home/kali/OS_LAB/umair_086/question2"` and its output: `Hello`, `Hello`, `Hello`, `Hi`, `Hi`, `Jello`, `Jello`, and `Polo Polo`. The prompt `kali@kali:~/OS_LAB/umair_086$` is visible at the bottom of the terminal.

Question 3:

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

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

```
struct pcb{
```

```
    int pid,burst_time,arrival_time,waiting_time,turnaround_time;
```

```
};
```

```
struct pcb processes[3];
```

```
struct pcb temp;
```

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

```
    for(int i=0;i<3;i++)
```

```
    {
```

```

printf("Enter the burst time of process %d\n",i+1);
scanf("%d",&processes[i].burst_time);
printf("Enter the arrival time of process %d\n",i+1);
scanf("%d",&processes[i].arrival_time);
processes[i].pid=i+1;

}

//sorting on the basis of arrival time
for(int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        if(processes[j].arrival_time > processes[j+1].arrival_time)
        {
            //swapping
            temp=processes[j];
            processes[j]=processes[j+1];
            processes[j+1]=temp;
        }
    }
}

int sum=0;

for(int i=0;i<3;i++)
{
    sum=sum+processes[i].burst_time;
    processes[i].turnaround_time=sum;
}

printf("pid\tburst_time \tarrival_time\twaiting_time\n");

```

```

for(int i=0;i<3;i++)
{
    processes[i].waiting_time=processes[i].turnaround_time-processes[i].burst_time;

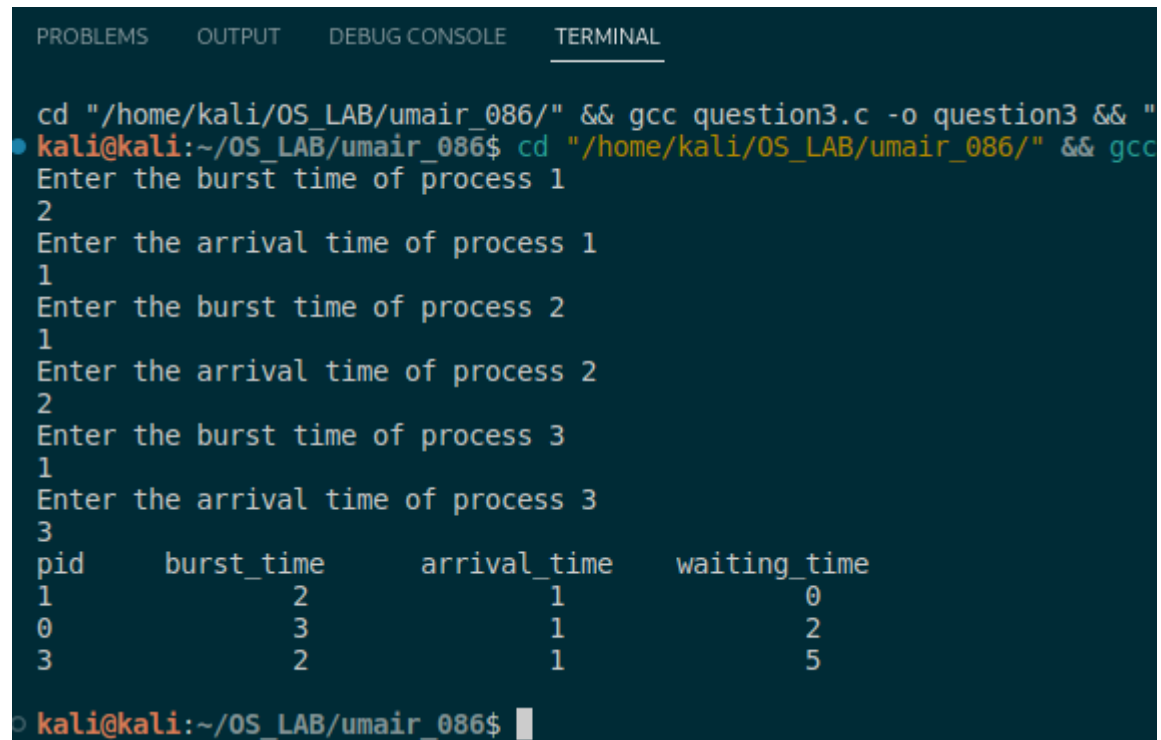
printf("%d\t%d\t%d\t%d\n",processes[i].pid,processes[i].burst_time,processes[i].arrival_time,
processes[i].waiting_time);

}

return 0;
}

```

Output:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/home/kali/OS_LAB/umair_086/" && gcc question3.c -o question3 && "
kali@kali:~/OS_LAB/umair_086$ cd "/home/kali/OS_LAB/umair_086/" && gcc
Enter the burst time of process 1
2
Enter the arrival time of process 1
1
Enter the burst time of process 2
1
Enter the arrival time of process 2
2
Enter the burst time of process 3
1
Enter the arrival time of process 3
3
pid      burst_time    arrival_time    waiting_time
1         2             1              0
0         3             1              2
3         2             1              5
kali@kali:~/OS_LAB/umair_086$

```