**FA20-BSE-086**

**UMAIR ASAD**

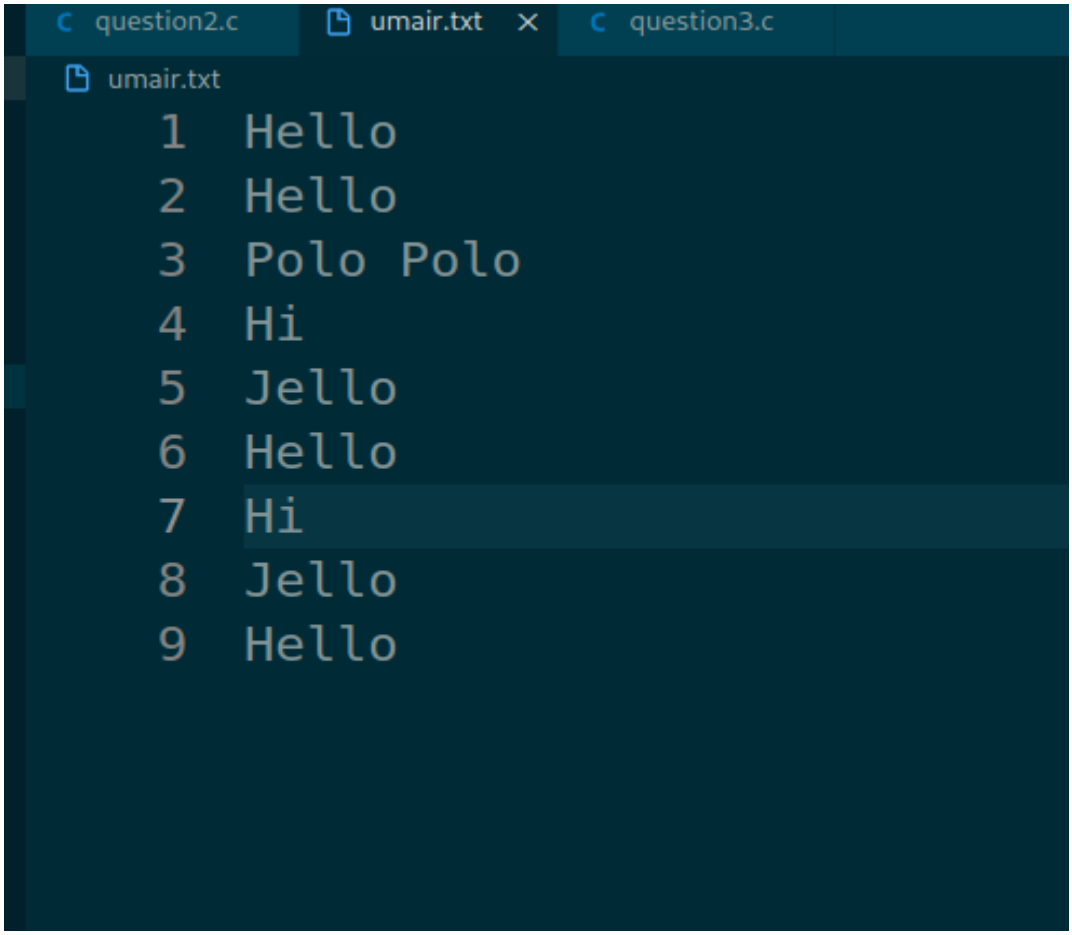
**MID TERM EXAM**

**OS LAB**

**SEC A**

## Question 1:

Umair.txt

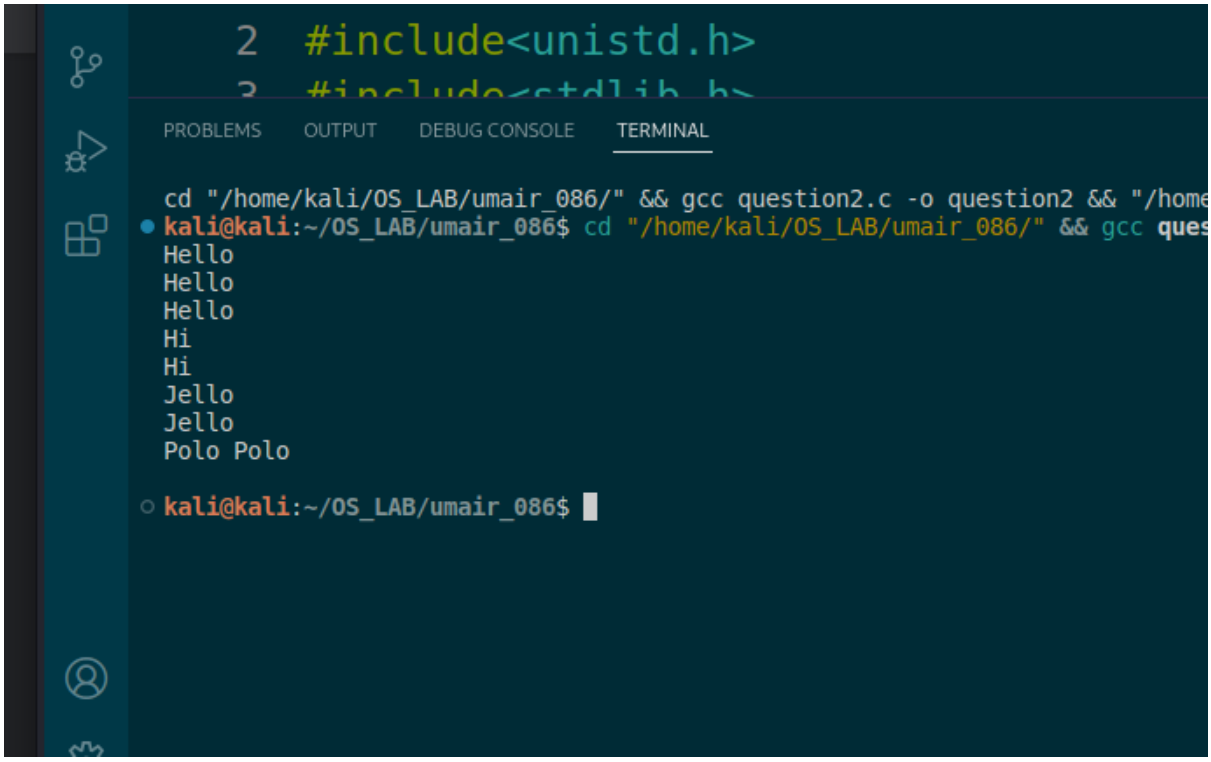



## Question 2:

```
#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:




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


    }
    return 0;
}
```

## Output: