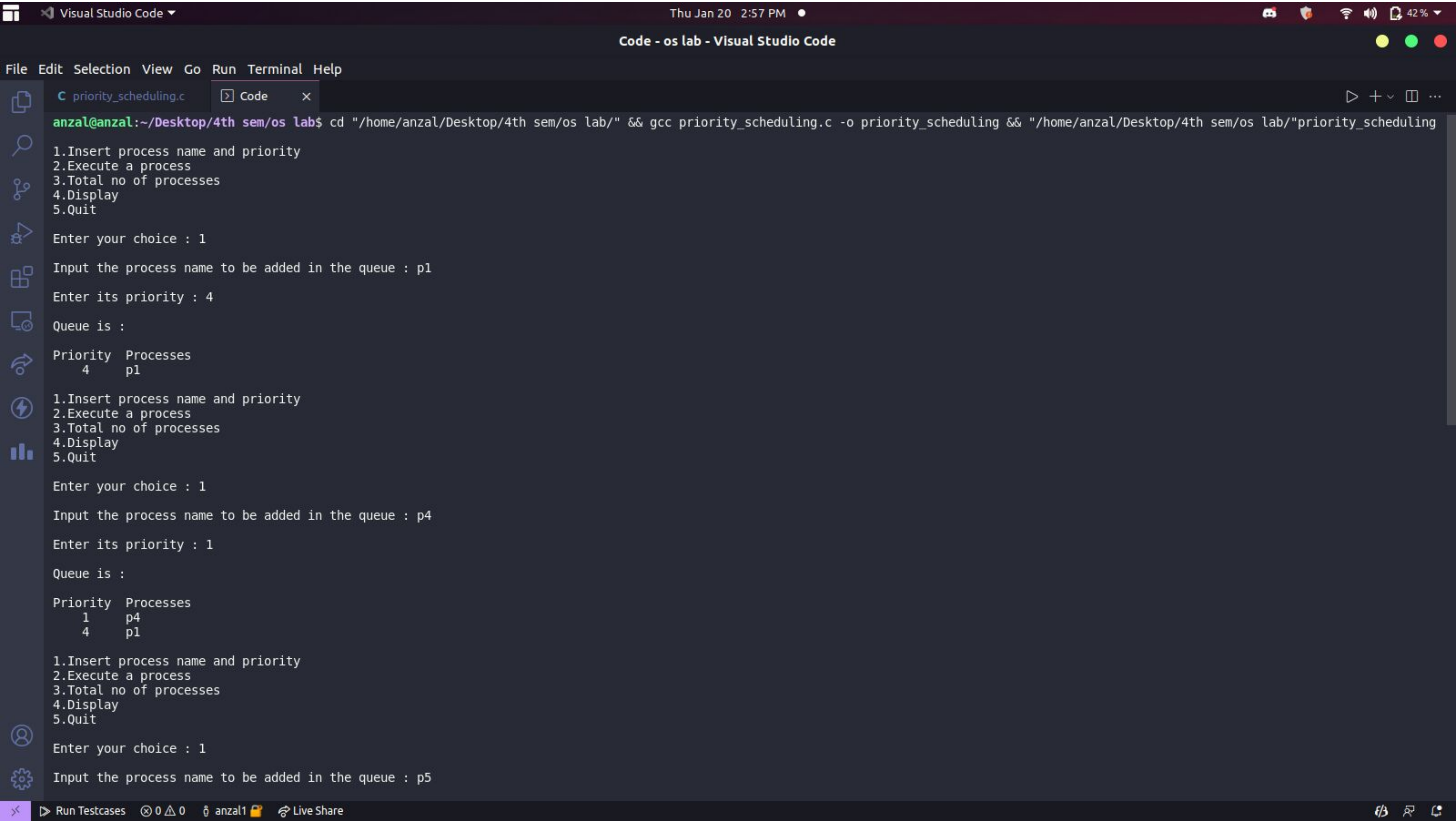
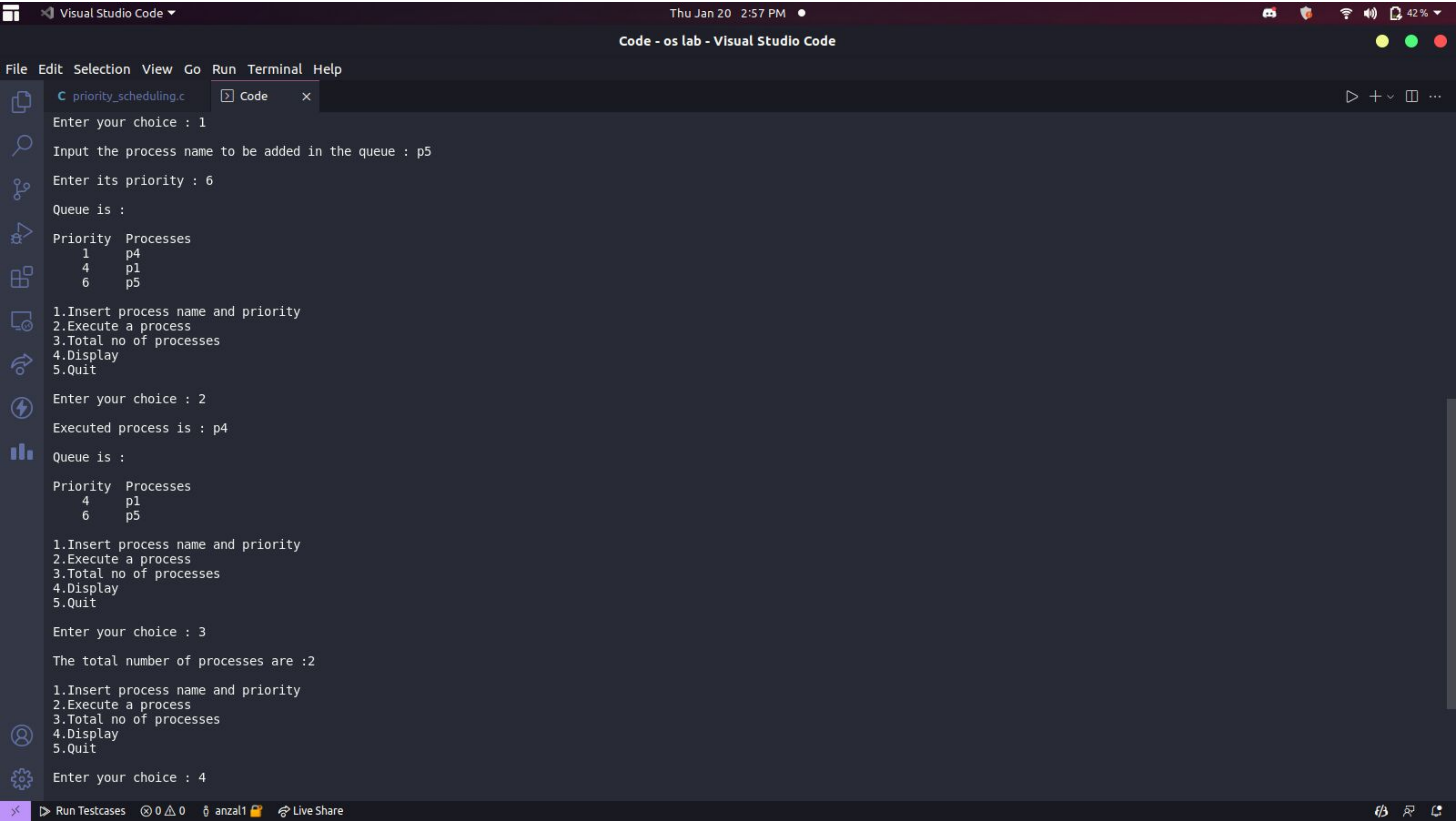
ANZAL HUSAIN ABIDI 20BCS009

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int i = 0;
struct node
{
    int priority;
    char name[100];
    struct node *link;
} *front = NULL;
void insert();
void del();
void display();
int isEmpty();
void front_and_rear();
int main()
{
    int choice, item, item_priority;
    while (1)
    {
        printf("\n1.Insert process name and priority\n");
        printf("2.Execute a process\n");
        printf("3.Total no of processes\n");
        printf("4.Display\n");
        printf("5.Quit\n");
        printf("\nEnter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            insert();
            display();
            i++;
            break;
        case 2:
            del();
            display();
            i--;
            break;
        case 3:
            printf("\nThe total number of processes are :%d\n", i);
            break;
        case 4:
            display();
            break;
        case 5:
            exit(1);
        default:
            printf("\nWrong choice\n");
        }
    }
    return 0;
```

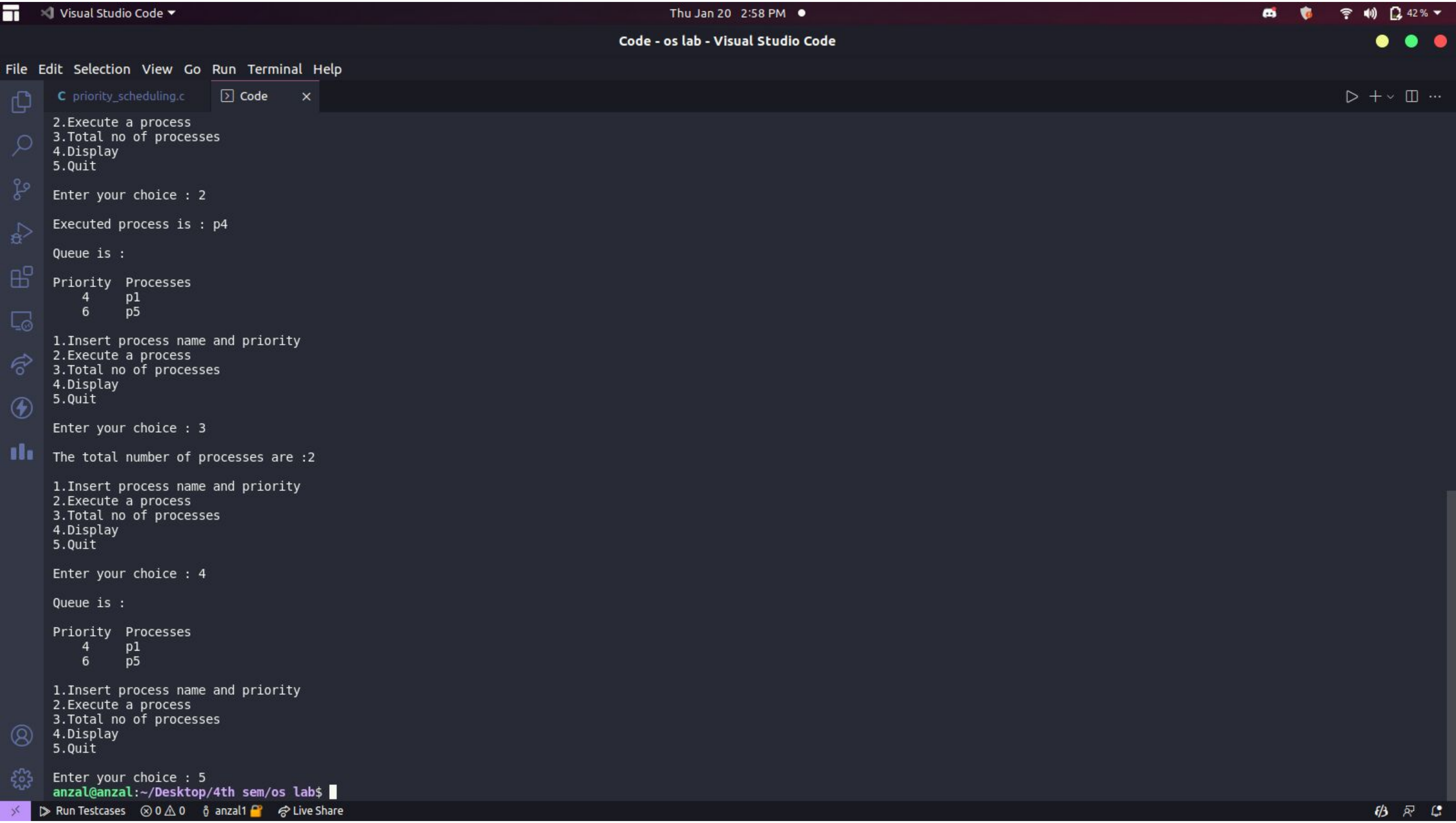
```
}
void insert()
{
    struct node *tmp, *p;
    tmp = (struct node *)malloc(sizeof(struct node));
    if (tmp == NULL)
    {
        printf("\nMemory not available\n");
        return;
    }
    printf("\nInput the process name to be added in the queue : ");
    scanf(" %[^\n]s", tmp->name);
    printf("\nEnter its priority : ");
    scanf("%d", &(tmp->priority));
    int item_priority = tmp->priority;
    if (isEmpty() || item_priority < front->priority)
    {
        tmp->link = front;
        front = tmp;
    }
    else
    {
        p = front;
        while (p->link != NULL && p->link->priority <= item_priority)
            p = p->link;
        tmp->link = p->link;
        p->link = tmp;
    }
}
void del()
{
    struct node *tmp;
    char item[100];
    if (isEmpty())
    {
        printf("\nQueue Underflow\n");
        exit(1);
    }
    else
    {
        tmp = front;
        strcpy(item, tmp->name);
        front = front->link;
        free(tmp);
    }
    printf("\nExecuted process is : %s\n", item);
}
int isEmpty()
{
    if (front == NULL)
        return 1;
    else
        return 0;
}
void display()
{
    struct node *ptr;
    ptr = front;
    if (isEmpty())
        printf("\nQueue is empty\n");
    else
    {
        printf("\nQueue is :\n");
printf("\nPriority  Processes\n");
while(ptr!=NULL)
{
printf("%5d  %5s\n",ptr->priority,ptr->name);
ptr=ptr->link;
}
    }
}
```



```
anzal@anzal:~/Desktop/4th sem/os lab$ cd "/home/anzal/Desktop/4th sem/os lab/" && gcc priority_scheduling.c -o priority_scheduling && "/home/anzal/Desktop/4th sem/os lab/"priority_scheduling

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 1

Input the process name to be added in the queue : p1

Enter its priority : 4

Queue is :

Priority  Processes
    4     p1

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 1

Input the process name to be added in the queue : p4

Enter its priority : 1

Queue is :

Priority  Processes
    1     p4
    4     p1

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 1

Input the process name to be added in the queue : p5
```



```
Enter your choice : 1

Input the process name to be added in the queue : p5

Enter its priority : 6

Queue is :

Priority  Processes
    1     p4
    4     p1
    6     p5

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 2

Executed process is : p4

Queue is :

Priority  Processes
    4     p1
    6     p5

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 3

The total number of processes are :2

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 4
```



```
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 2

Executed process is : p4

Queue is :

Priority  Processes
    4     p1
    6     p5

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 3

The total number of processes are :2

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 4

Queue is :

Priority  Processes
    4     p1
    6     p5

1.Insert process name and priority
2.Execute a process
3.Total no of processes
4.Display
5.Quit

Enter your choice : 5
anzal@anzal:~/Desktop/4th sem/os lab$ 
```