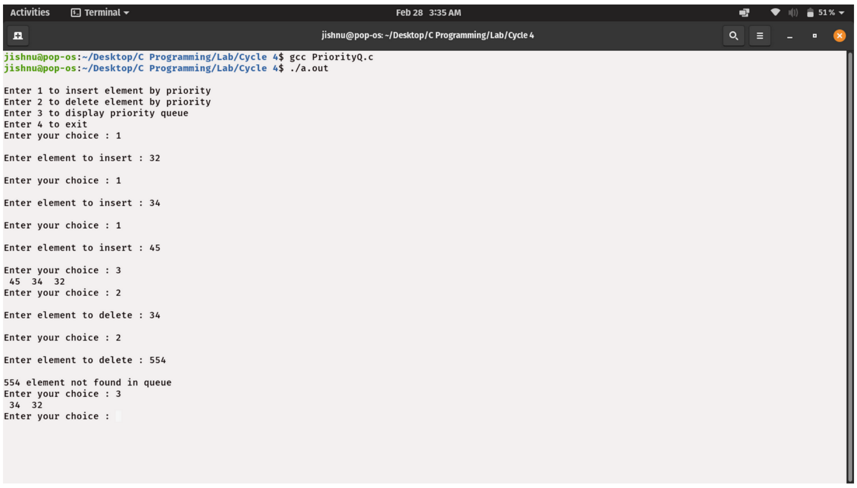
(1) PRIORITY QUEUE

```
#include<stdio.h>
#include <stdlib.h>
#define MAX 10

void create_queue();
void insert_element(int);
void delete_element(int);
void check_priority(int);
void display_priorityqueue();

int pqueue[MAX];
int front, rear;

void main()
{
    int n, choice;
    printf("\nEnter 1 to insert element by priority ");
    printf("\nEnter 2 to delete element by priority ");
    printf("\nEnter 3 to display priority queue ");
    printf("\nEnter 4 to exit");
    create_queue();
    while (1)
    {
        printf("\nEnter your choice : ");
        scanf("%d", &choice);
        switch(choice)
        {
        case 1:
            printf("\nEnter element to insert : ");
            scanf("%d",&n);
            insert_element(n);
            break;
        case 2:
            printf("\nEnter element to delete : ");
            scanf("%d",&n);
            delete_element(n);
            break;
        case 3:
            display_priorityqueue();
            break;
        case 4:
            exit(0);
        default:
            printf("\n Please enter valid choice");
        }
    }
}
void create_queue() {
    front = rear = -1;
}
void insert_element(int data) {
    if (rear >= MAX - 1) {
        printf("\nQUEUE OVERFLOW");
        return;
    }else if ((front == -1) && (rear == -1)){
        front++;
        rear++;
        pqueue[rear] = data;
        return;
    }else{
        check_priority(data);
    rear++;
}}
void check_priority(int data) {
    int i,j;
    for (i = 0; i <= rear; i++){
        if (data >= pqueue[i]){
            for (j = rear + 1; j > i; j--){
                pqueue[j] = pqueue[j - 1];
            }
            pqueue[i] = data;
            return;
        }
    }
    pqueue[i] = data;
}
void delete_element(int data){
    int i;
    if ((front==-1) && (rear==-1)){
        printf("\nEmpty Queue");
        return;
    }
    for (i = 0; i <= rear; i++){
        if (data == pqueue[i]){
            for (i=0; i < rear; i++) {
                pqueue[i] = pqueue[i + 1];
            }
            pqueue[i] = -99;
            rear--;
            if (rear == -1)
              front = -1;
            return;
        }
    }
    printf("\n%d element not found in queue", data);
}
void display_priorityqueue() {
    if ((front == -1) && (rear == -1)){
        printf("\nEmpty Queue ");
        return;
    }
    for (front =0 ; front <= rear; front++){
        printf(" %d ", pqueue[front]);
    }
     front = 0;
}
```

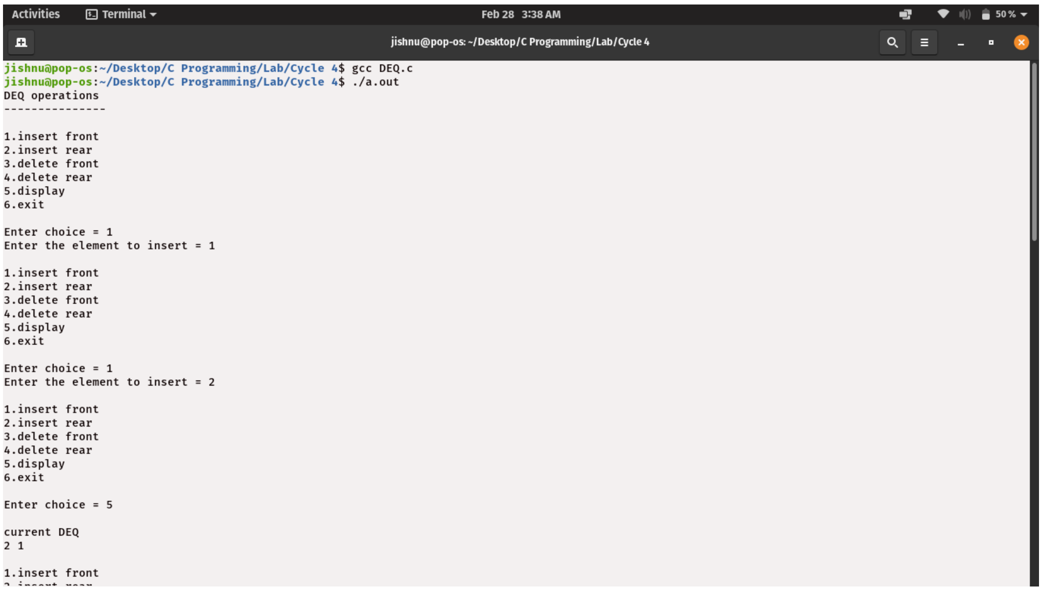


```
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 4$ gcc PriorityQ.c
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 4$ ./a.out

Enter 1 to insert element by priority
Enter 2 to delete element by priority
Enter 3 to display priority queue
Enter 4 to exit
Enter your choice : 1

Enter element to insert : 32

Enter your choice : 1

Enter element to insert : 34

Enter your choice : 1

Enter element to insert : 45

Enter your choice : 3
 45  34  32
Enter your choice : 2

Enter element to delete : 34

Enter your choice : 2

Enter element to delete : 554

554 element not found in queue
Enter your choice : 3
 34  32
Enter your choice :
```

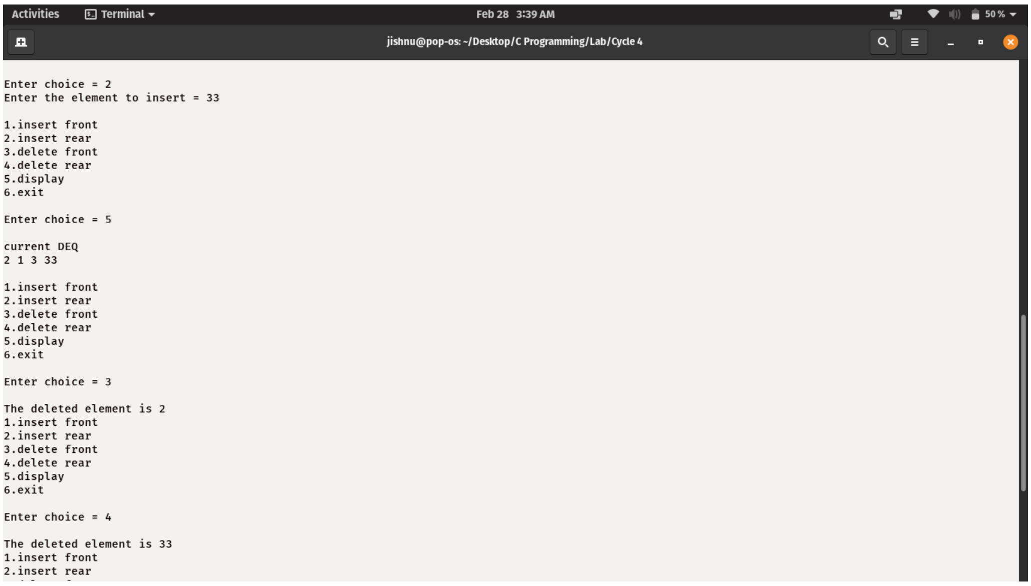
(2) DOUBLE ENDED QUEUE

```
#include <stdio.h>
#include<stdlib.h>
#define max 5
int DQ[max];
int front =-1, rear = -1, item;
void insert_front(){
	if((front==0 && rear==max-1) || (front==rear+1)){
		printf("Overflow");
	}else if((front==-1) && (rear==-1)){
		front=rear=0;
		printf("Enter the element to insert = ");
		scanf("%d",&item);
		DQ[front]=item;
	}else if(front==0){
		front=max-1;
		printf("Enter the element to insert = ");
		scanf("%d",&item);
		DQ[front]=item;

	}else{
		front=front-1;
		printf("Enter the element to insert = ");
		scanf("%d",&item);
		DQ[front]=item;
	}
}
void insert_rear(){
	if((front==0 && rear==max-1) || (front==rear+1)){
		printf("Overflow");
	}else if((front==-1) && (rear==-1)){
		rear=0;
		printf("Enter the element to insert = ");
		scanf("%d",&item);
		DQ[rear]=item;
	}else if(rear==max-1){
		rear=0;
		DQ[rear]=item;
	}else{
		rear++;
		printf("Enter the element to insert = ");
		scanf("%d",&item);
		DQ[rear]=item;
	}
}

void display(){
	int i=front;
	printf("\ncurrent DEQ\n");
	while(i != rear){
		printf("%d ",DQ[i]);
		i=(i+1)%max;
	}
	printf("%d\n",DQ[rear]);
}

void delete_front(){
	if((front==-1) && (rear==-1)){
		printf("DEQ underflow");
	}else if(front==rear){
```

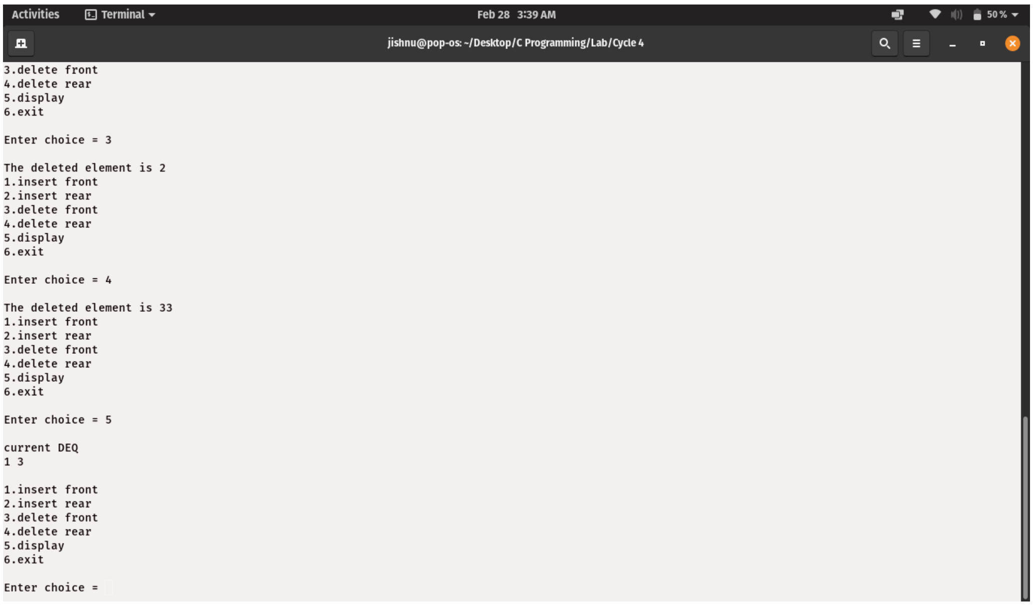
```
                printf("\nThe deleted element is %d", DQ[front]);
                front=-1;
                rear=-1;
        }else if(front==(max-1)){
                printf("\nThe deleted element is %d", DQ[front]);
                front=0;
        }else{
                printf("\nThe deleted element is %d", DQ[front]);
                front=front+1;
        }
}

void delete_rear(){
        if((front==-1) && (rear==-1)){
                printf("DEQ underflow");
        }else if(front==rear){
                printf("\nThe deleted element is %d", DQ[rear]);
                front=-1;
                rear=-1;
        }else if(rear==0){
                printf("\nThe deleted element is %d", DQ[rear]);
                rear=max-1;
        }else{
                printf("\nThe deleted element is %d", DQ[rear]);
                rear=rear-1;
        }
}

void main(){
        printf("DEQ operations\n");
        for(int k=0;k<15;k++){
                printf("%c",'-');
        }printf("\n");
        int op = 1;
        while(op<7){
                printf("\n1.insert front\n2.insert rear\n3.delete front\n4.delete rear\n5.display\n6.exit\n\n");
                printf("Enter choice = ");
                scanf("%d",&op);
                switch(op){
                case 1 : insert_front();
                        break;
                case 2 : insert_rear();
                        break;
                case 3 : delete_front();
                        break;
                case 4 : delete_rear();
                        break;
                case 5 : display();
                        break;
                case 6 : printf("exiting the program...\n");
                        exit(0);
                default: printf("something went wrong...program terminated\n");
                        exit(0);
                }
        }
}
```



```
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 4$ gcc DEQ.c
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 4$ ./a.out
DEQ operations
---------------

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 1
Enter the element to insert = 1

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 1
Enter the element to insert = 2

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 5

current DEQ
2 1

1.insert front
2.insert rear
```



```
Enter choice = 2
Enter the element to insert = 33

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 5

current DEQ
2 1 3 33

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 3

The deleted element is 2
1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 4

The deleted element is 33
1.insert front
2.insert rear
```



```
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 3

The deleted element is 2
1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 4

The deleted element is 33
1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 5

current DEQ
1 3

1.insert front
2.insert rear
3.delete front
4.delete rear
5.display
6.exit

Enter choice = 
```