ANVITA KUMAR
C-22
Roll No.: 2104097

//WAP to implement linear queue ADT using linked list

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
   int data;
   struct node *next;
};
struct queue
{
   struct node *front;
   struct node *rear;
};
struct node *front = NULL;
struct node *rear = NULL;
struct queue *q;
struct queue *enqueue(struct queue *, int);
struct queue *dequeue(struct queue *q);
int getFront(struct queue *);
int getRear(struct queue *);
int isEmpty();
struct queue *display(struct queue *);

int main()
{
   int val, ch;
   do
   {
      printf("\n*****List Of Operations*****\n");
```

ANVITA KUMAR
C-22
Roll No.: 2104097

```
    printf("1. ENQUEUE\n2. DEQUEUE\n3. GET FRONT\n4. GET REAR\n5. IS EMPTY\n6. DISPLAY\n7. EXIT\n");

    printf("Enter your choice: ");

    scanf("%d",&ch);

    switch(ch) {

    case 1:

      printf("Enter the value to be inserted in the queue: ");

      scanf("%d", &val);

      q = enqueue(q, val);

      break;

    case 2:

      q = dequeue(q);

      break;

    case 3:

      val = getFront(q);

      if (val != -1)

        printf("The front element is: %d\n", val);

      break;

    case 4:

      val = getRear(q);

      if (val != -1)

        printf("The rear element is: %d\n", val);

      break;

    case 5:

      isEmpty(q);

      break;

    case 6:

      q = display(q);

      break;

    case 7:

      printf("\tEXIT POINT!");
```

ANVITA KUMAR
C-22
Roll No.: 2104097

```
            break;
        }
    } while (ch != 7);
    return 0;
}
struct queue *enqueue(struct queue *q, int val)
{
    struct node *newNode = (struct node *)malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = NULL;
    if (isEmpty())
    {
        rear = newNode;
        front = rear;
    }
    else
    {
        rear->next = newNode;
        rear = rear->next;
    }
}
struct queue *dequeue(struct queue *q)
{
    if (isEmpty())
    {
        printf("UNDERFLOW\n");
        return q;
    }
    else
    {
        struct node *temp = front;
```

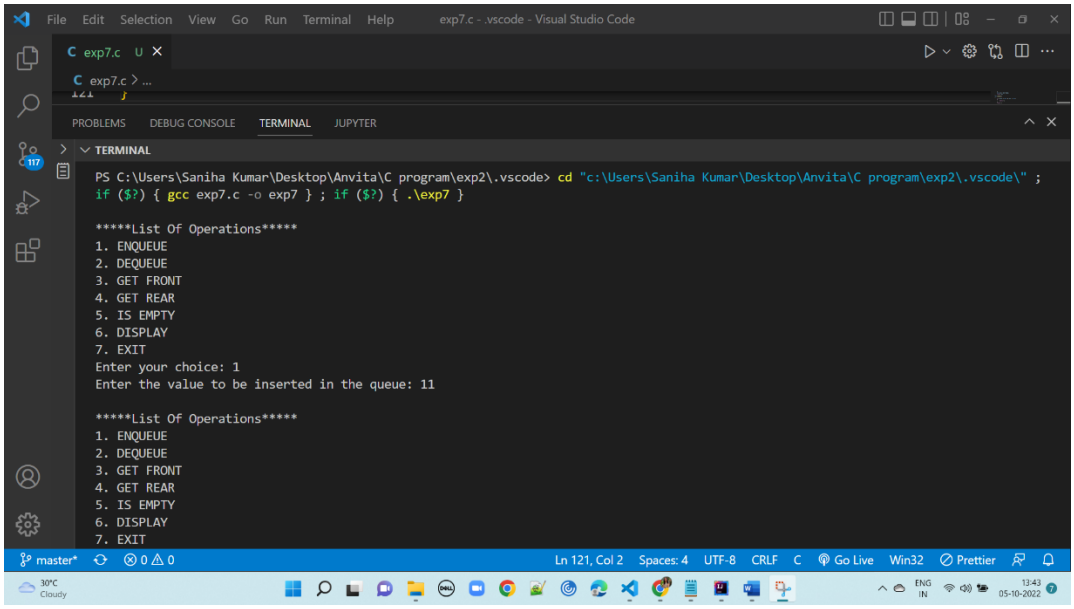
ANVITA KUMAR
C-22
Roll No.: 2104097

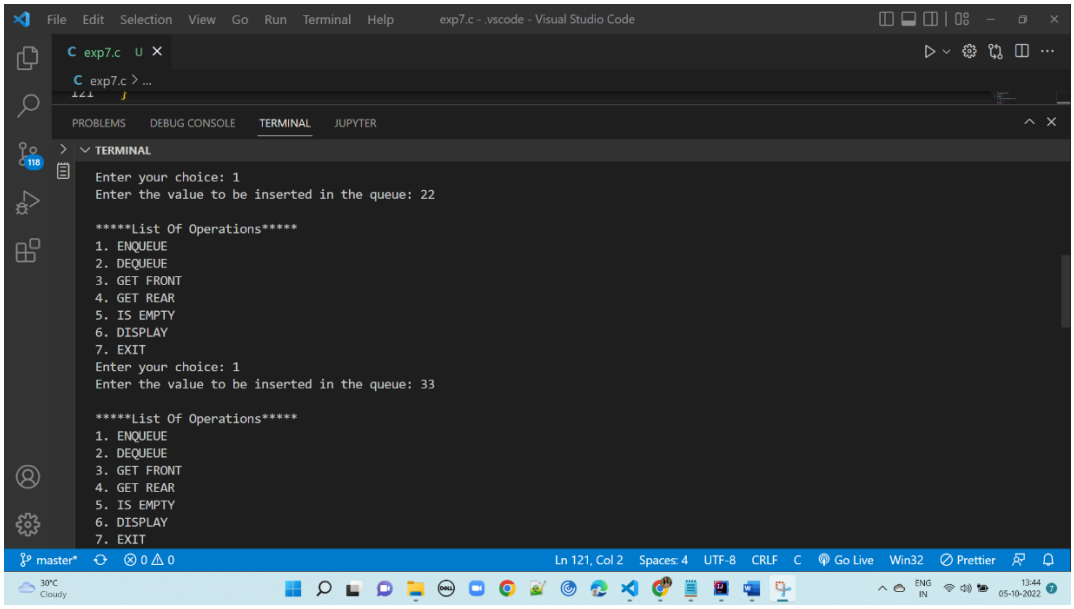
```
        front = front->next;
        printf("The value being deleted is : %d\n", temp->data);
        free(temp);
    }
}
int getFront(struct queue *q)
{
    if (isEmpty())
    {
        printf("QUEUE IS EMPTY\n");
        return -1;
    }
    int val = front->data;
    return val;
}
int getRear(struct queue *q)
{
    if (isEmpty())
    {
        printf("QUEUE IS EMPTY\n");
        return -1;
    }
    int val = rear->data;
    return val;
}
int isEmpty()
{
    if (front == NULL && rear == NULL)
    {
        return -1;
    }
```

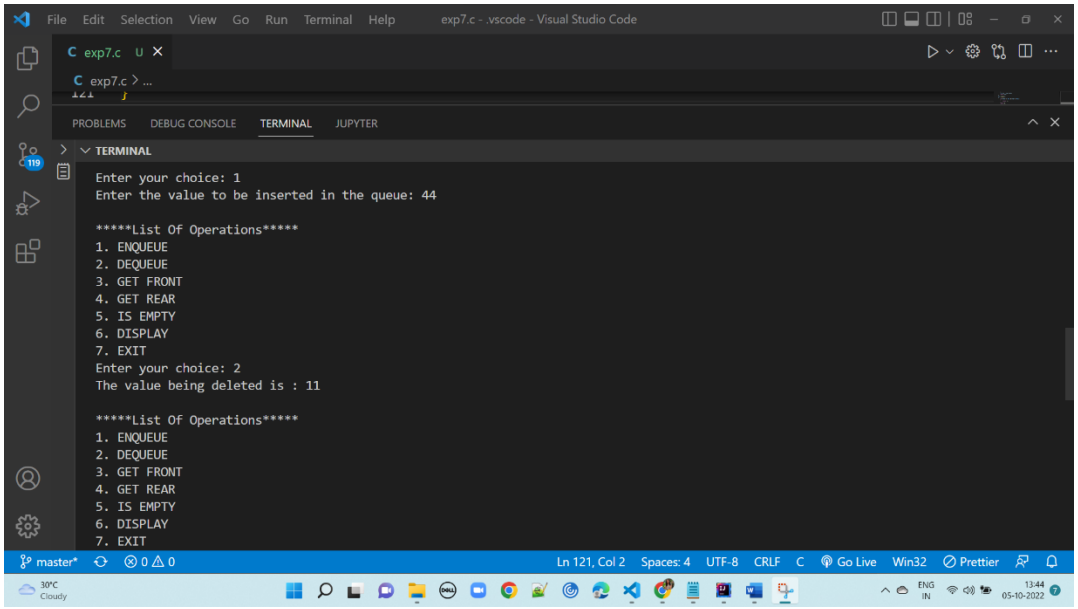
ANVITA KUMAR
C-22
Roll No.: 2104097

```
    return 0;
}
struct queue *display(struct queue *q)
{
    if (isEmpty())
    {
        printf("QUEUE IS EMPTY\n");
        return q;
    }
    struct node *temp = front;
    printf("The Queue is: ");
    while (temp != NULL)
    {
        printf("\t%d", temp->data);
        temp = temp->next;
    }
    printf("\tNULL\n");
}
```

ANVITA KUMAR
C-22
Roll No.: 2104097







ANVITA KUMAR
C-22
Roll No.: 2104097