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 node *front = NULL;

struct node *rear = NULL;

int main()

{

   int val, ch;

   do

   {

      printf("\n*****List Of Operations*****\n");

      printf("1. ENQUEUE\n2. DEQUEUE\n3. GET FRONT\n4. GET REAR\n5. DISPLAY\n6. 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);

         enqueue(val);

         break;

      case 2:

         dequeue();

         break;

      case 3:

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

         break;
```

ANVITA KUMAR
C-22
Roll No.: 2104097

```
        case 4:
            printf("The rear element is: %d\n", getRear());
            break;
        case 5:
            display();
            break;
        case 6:
            printf("\tEXIT POINT!");
            break;
        }
    } while (ch != 6);
    return 0;
}
int isEmpty()
{
    if (front == NULL && rear == NULL)
    {
        return 1;
    }
    return 0;
}
void enqueue(int val)
{
    struct node *newNode = (struct node *)malloc(sizeof(struct node));
    newNode->data = val;
    newNode->next = NULL;
    if (isEmpty())
    {
        rear = newNode;
        front = rear;
    }
```

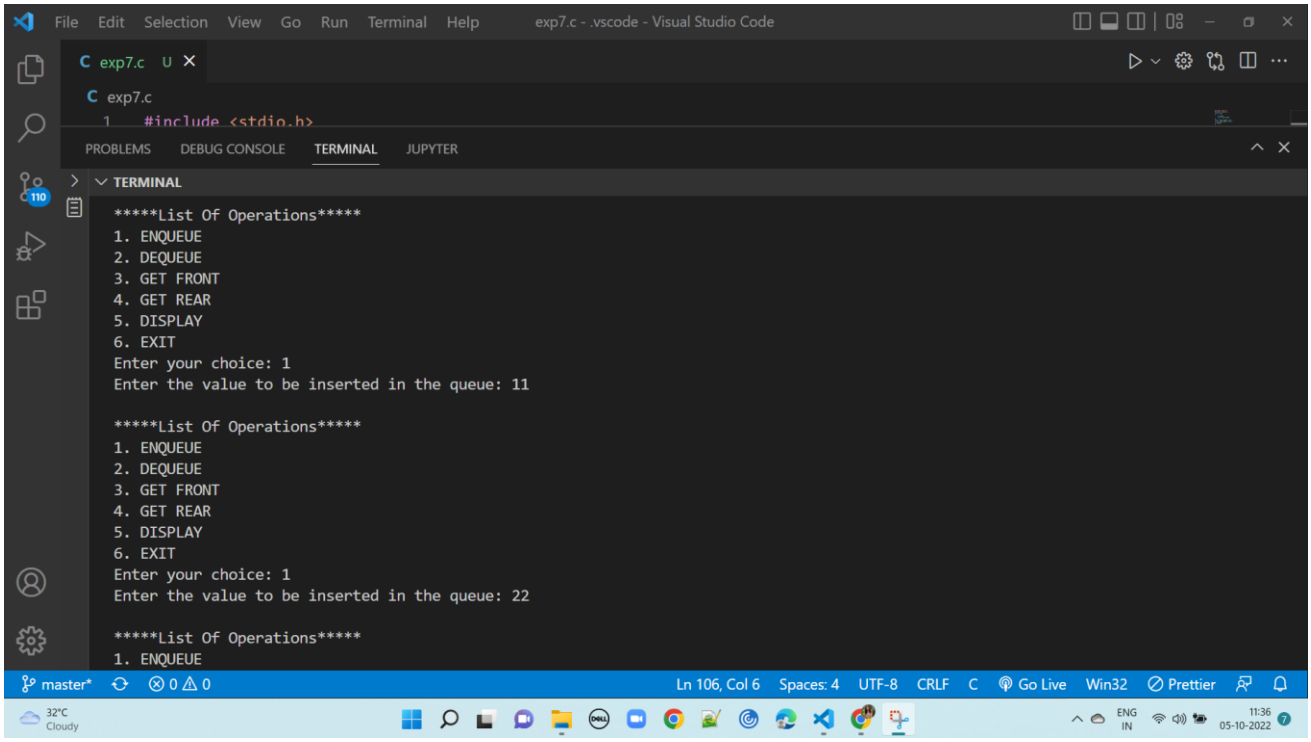
ANVITA KUMAR
C-22
Roll No.: 2104097

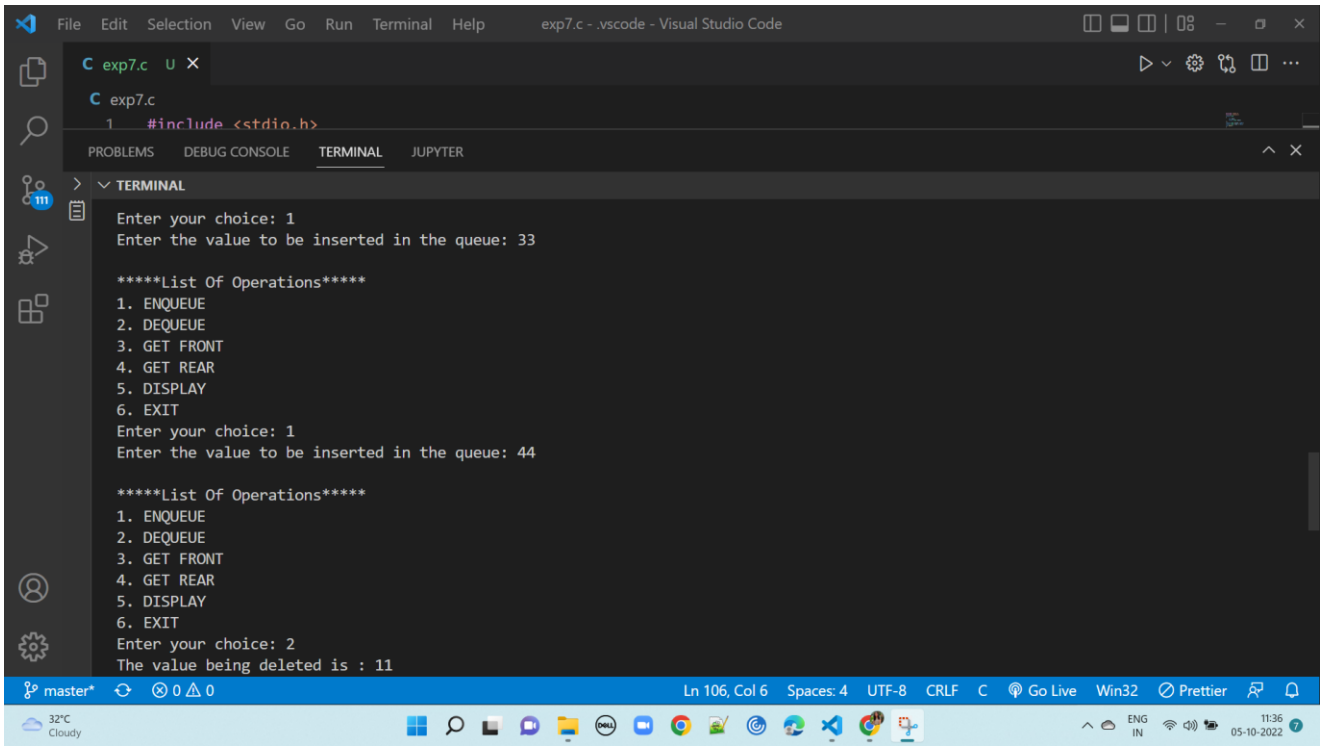
```
    else
    {
        rear->next = newNode;
        rear = rear->next;
    }
}
void dequeue()
{
    if (isEmpty())
    {
        printf("UNDERFLOW\n");
        return;
    }
    else
    {
        struct node *temp = front;
        front = front->next;
        printf("The value being deleted is : %d\n", temp->data);
        free(temp);
    }
}
int getFront()
{
    if (isEmpty())
    {
        printf("QUEUE IS EMPTY\n");
        return -1;
    }
    int val = front->data;
    return val;
}
```

ANVITA KUMAR
C-22
Roll No.: 2104097

```
int getRear()
{
   if (isEmpty())
   {
      printf("QUEUE IS EMPTY\n");
      return -1;
   }
   int val = rear->data;
   return val;
}
void display()
{
   if (isEmpty())
   {
      printf("QUEUE IS EMPTY\n");
      return;
   }
   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