NAME: NJIDDA SALIFU
MATRICULE: FE21A272

QUEUE IMPLEMENTATION SING LINKED LIST

CODE

```
// Name: NJIDDA SALIFU
// Matricule: FE21A272
// Implementation of Queues using linked list

#include <stdio.h>
#include <stdlib.h>
struct node{
int info;
struct node *ptr;
}*front,*rear,*temp,*front1;
int TopofQueue();
void enq(int data);
void deq();
void display();
void create();
int count = 0;
void main(){
int no, ch, e;
printf("Following are the operations performed on the Queue\n");
```

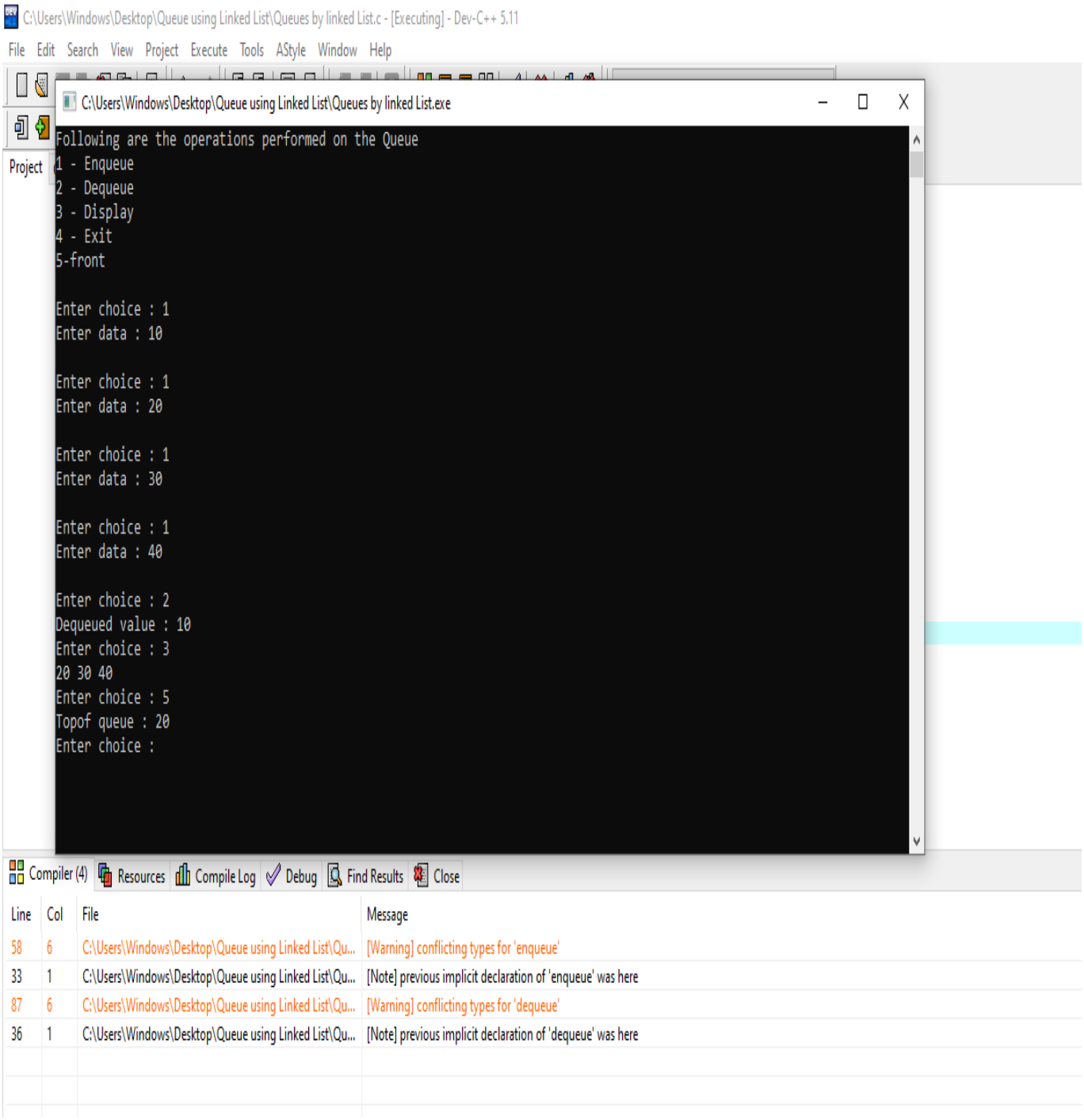
```
printf("1 - Enqueue\n");
printf("2 - Dequeue\n");
printf("3 - Display\n");
printf("4 - Exit\n");
printf("5-front\n");
while (1){
printf("\nEnter choice : ");
scanf("%d", &ch);
switch (ch){
case 1:
printf("Enter data : ");
scanf("%d", &no);
enqueue(no);
break;
case 2:
dequeue();
break;
case 3:
display();
break;
case 4:
exit(0);
break;
case 5:
```

```
e =TopofQueue();
if (e != 0)
printf("Topof queue : %d", e);
else
printf( "No front element in Queue");
break;
default:
printf("Wrong choice, Try again ");
break;
}
}
}
//This function adds data in the queue
void enqueue(int data){
if (rear == NULL){
rear = (struct node *)malloc(1*sizeof(struct node));
rear->ptr = NULL;
rear->info = data;
front = rear;
}else{
temp=(struct node *)malloc(1*sizeof(struct node));
rear->ptr = temp;
temp->info = data;
temp->ptr = NULL;
```

```
rear = temp;
}
count++;
}
//This function displays data in the queue
void display(){
front1 = front;
if ((front1 == NULL) && (rear == NULL)){
printf("Queue is empty");
return;
}
while (front1 != rear){
printf("%d ", front1->info);
front1 = front1->ptr;
}
if (front1 == rear)
printf("%d", front1->info);
}
void dequeue(){
front1 = front;
if (front1 == NULL){
printf("Error");
return;
}
```

```
else
if (front1->ptr != NULL){
front1 = front1->ptr;
printf("Dequeued value : %d", front->info);
free(front);
front = front1;
}else{
printf("Dequeued value : %d", front->info);
free(front);
front = NULL;
rear = NULL;
}
count--;
}
int TopofQueue(){
//This function gives the first element of the queue
if ((front != NULL) && (rear != NULL))
return(front->info);
else
return 0;
}
```

Output:


C:\Users\Windows\Desktop\Queue using Linked List\Queues by linked List.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
C:\Users\Windows\Desktop\Queue using Linked List\Queues by linked List.exe
Following are the operations performed on the Queue
1 - Enqueue
2 - Dequeue
3 - Display
4 - Exit
5-front
Enter choice : 1
Enter data : 10
Enter choice : 1
Enter data : 20
Enter choice : 1
Enter data : 30
Enter choice : 1
Enter data : 40
Enter choice : 2
Dequeued value : 10
Enter choice : 3
20 30 40
Enter choice : 5
Topof queue : 20
Enter choice :
Project
Compiler (4) Resources Compile Log Debug Find Results Close
Line Col File Message
58 6 C:\Users\Windows\Desktop\Queue using Linked List\Qu... [Warning] conflicting types for 'enqueue'
33 1 C:\Users\Windows\Desktop\Queue using Linked List\Qu... [Note] previous implicit declaration of 'enqueue' was here
87 6 C:\Users\Windows\Desktop\Queue using Linked List\Qu... [Warning] conflicting types for 'dequeue'
36 1 C:\Users\Windows\Desktop\Queue using Linked List\Qu... [Note] previous implicit declaration of 'dequeue' was here