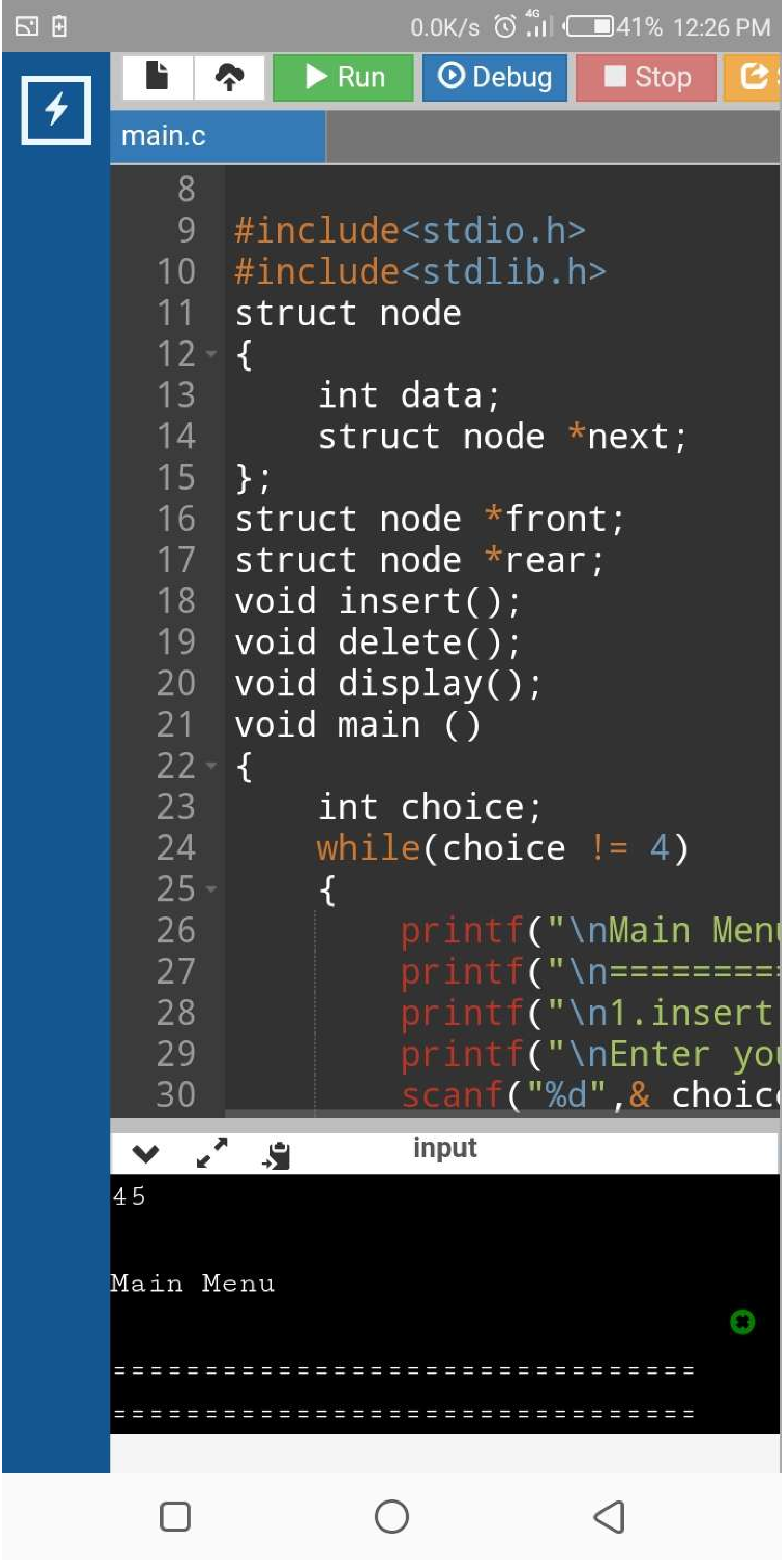
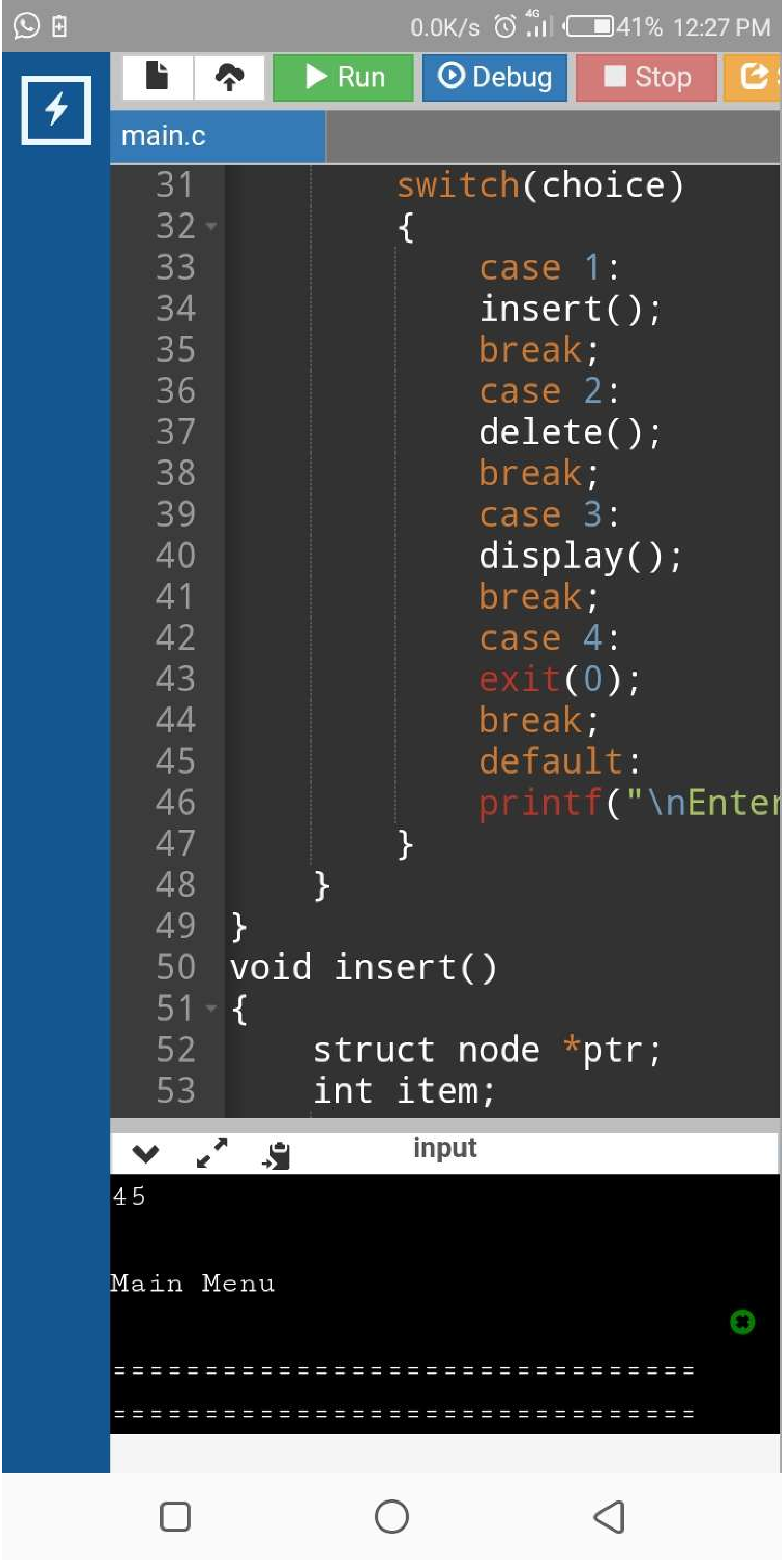


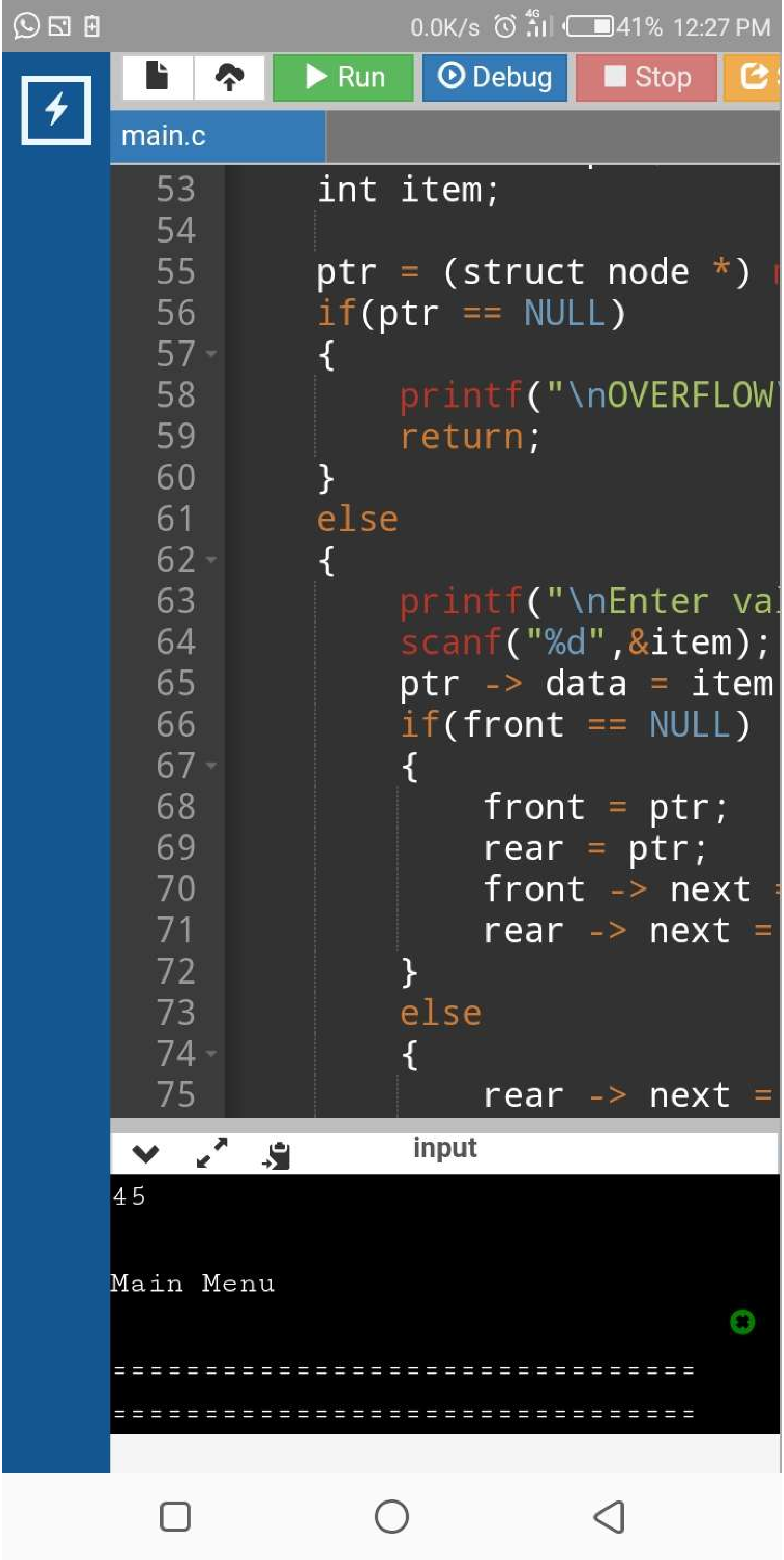
```c

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *front;
struct node *rear;
void insert();
void delete();
void display();
void main ()
{
    int choice;
    while(choice != 4)
    {
        printf("\nMain Menu
        printf("\n========
        printf("\n1.insert
        printf("\nEnter you
        scanf("%d",& choic
```

```
45


Main Menu


================================

================================
```



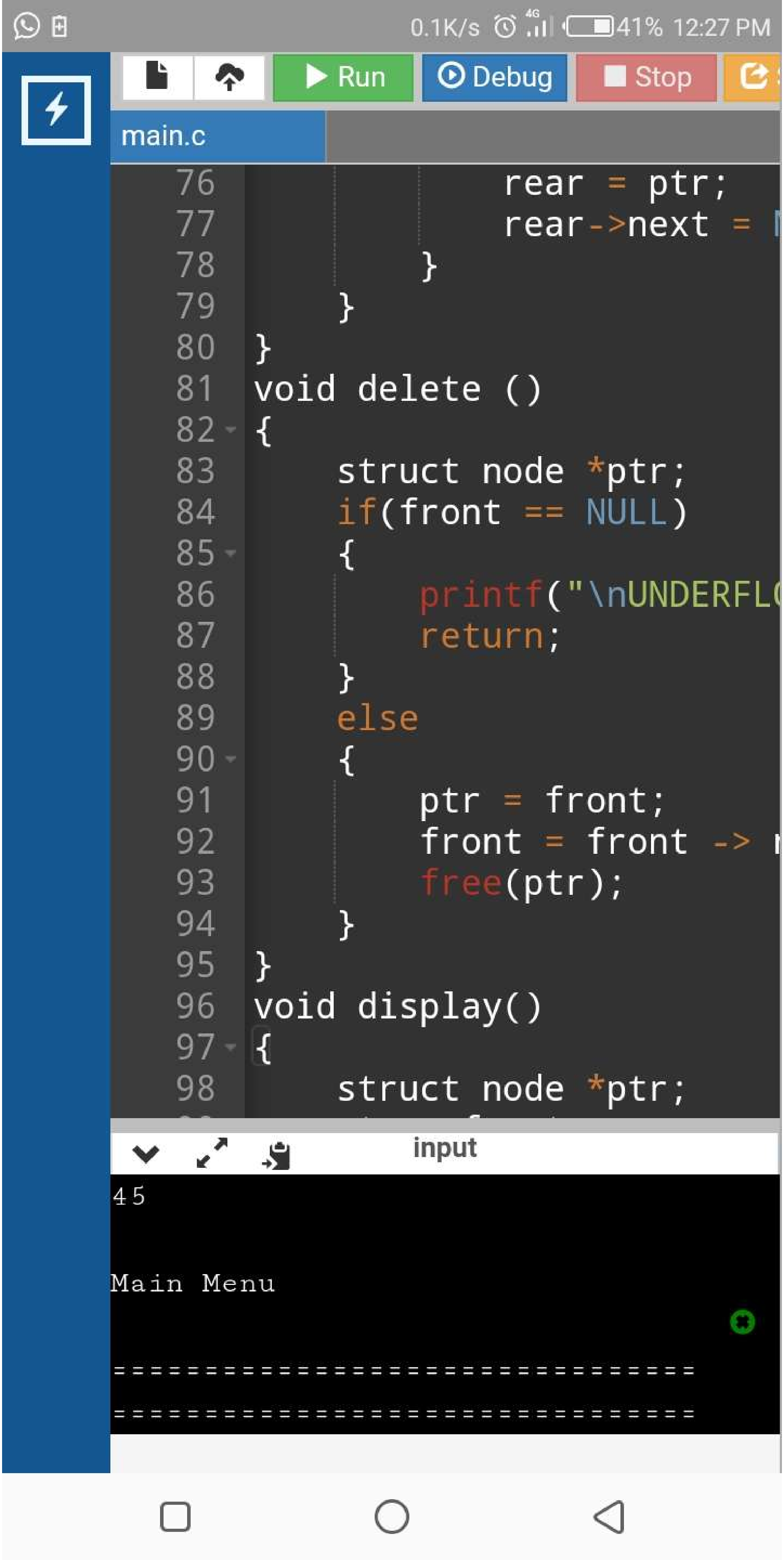
```c
        switch(choice)
        {
            case 1:
            insert();
            break;
            case 2:
            delete();
            break;
            case 3:
            display();
            break;
            case 4:
            exit(0);
            break;
            default:
            printf("\nEnter
        }
    }
}
void insert()
{
    struct node *ptr;
    int item;
```

```
45


Main Menu


================================
================================
```



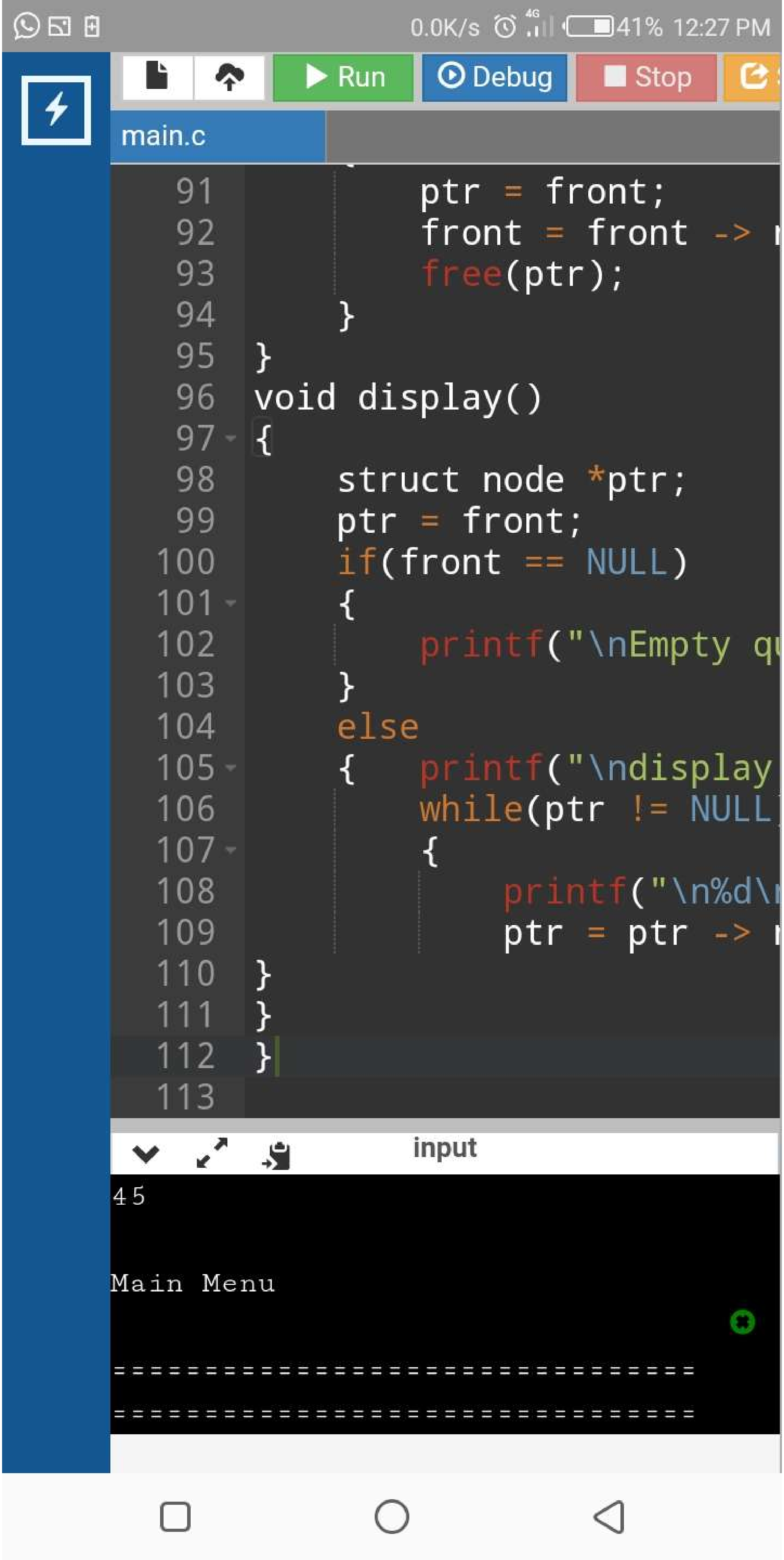
```c
    int item;

    ptr = (struct node *) 
    if(ptr == NULL)
    {
        printf("\nOVERFLOW
        return;
    }
    else
    {
        printf("\nEnter va
        scanf("%d",&item);
        ptr -> data = item
        if(front == NULL)
        {
            front = ptr;
            rear = ptr;
            front -> next 
            rear -> next =
        }
        else
        {
            rear -> next =
```

```
45


Main Menu


===================================
===================================
```



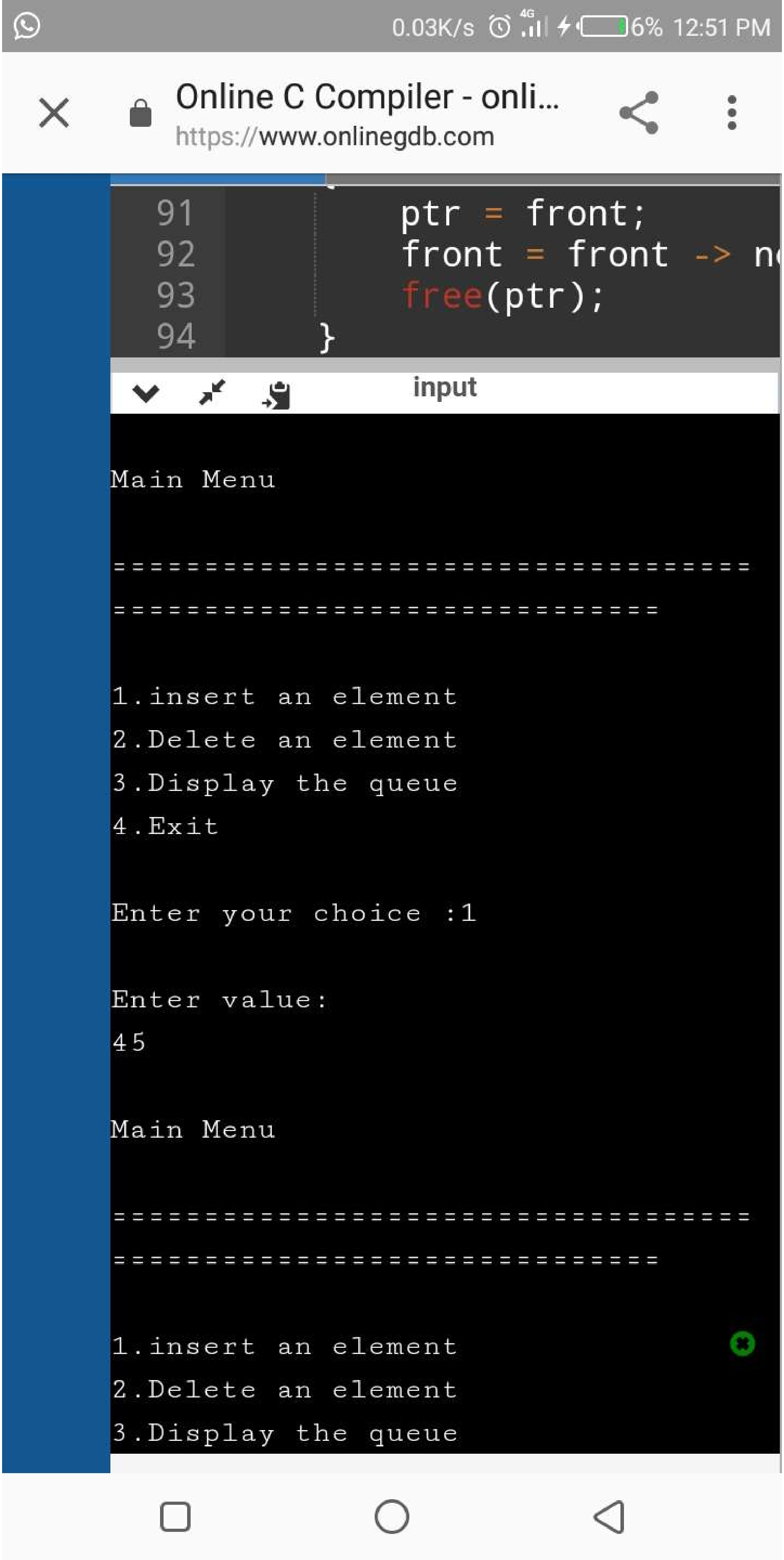
```
            rear = ptr;
            rear->next = N
        }
    }
}
void delete ()
{
    struct node *ptr;
    if(front == NULL)
    {
        printf("\nUNDERFL
        return;
    }
    else
    {
        ptr = front;
        front = front -> 
        free(ptr);
    }
}
void display()
{
    struct node *ptr;
```

```
4 5


Main Menu


================================
================================
```



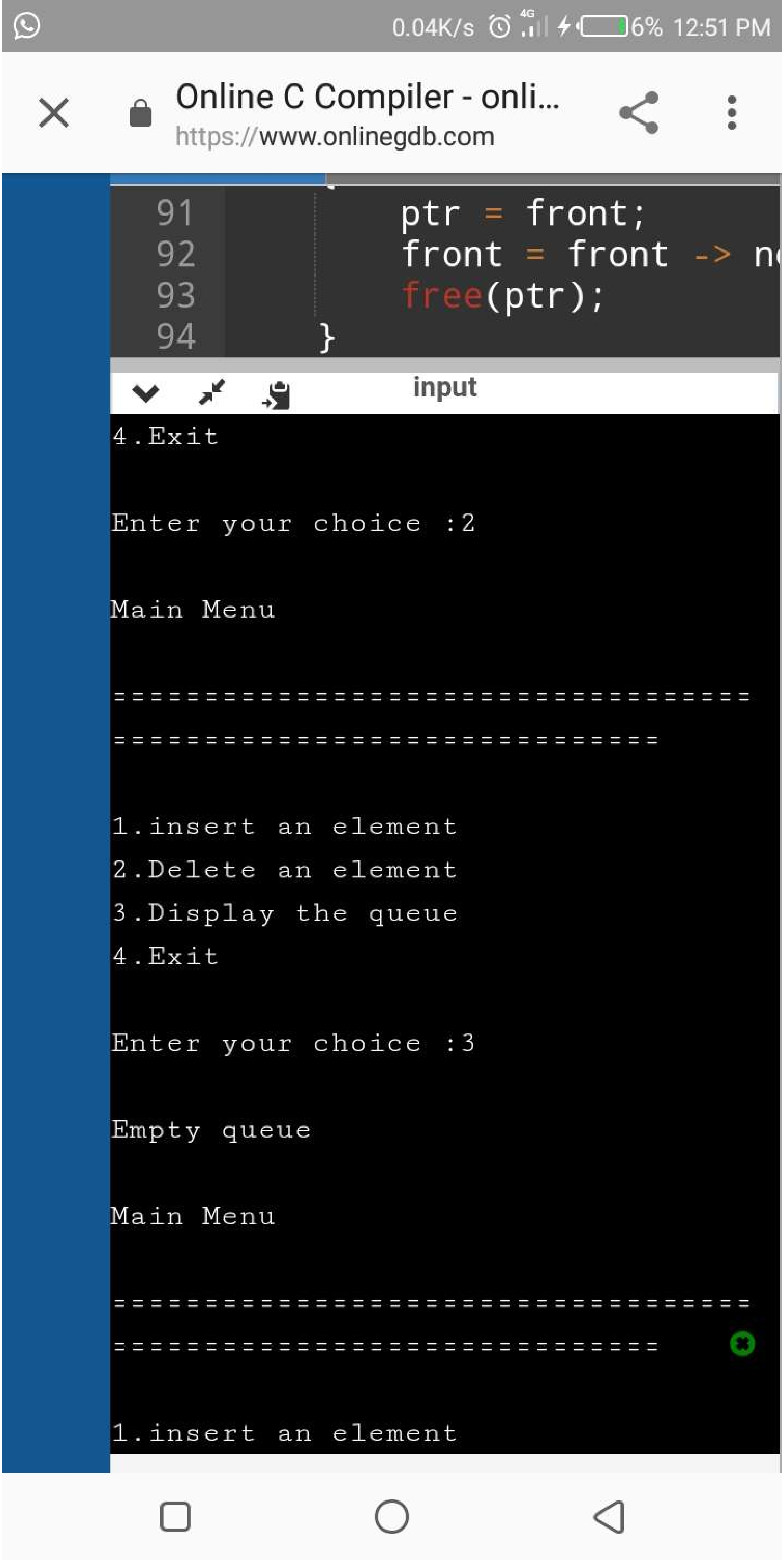
```c
        ptr = front;
        front = front -> r
        free(ptr);
    }
}
void display()
{
    struct node *ptr;
    ptr = front;
    if(front == NULL)
    {
        printf("\nEmpty qu
    }
    else
    {   printf("\ndisplay
        while(ptr != NULL
        {
            printf("\n%d\r
            ptr = ptr -> r
}
}
}
```

```
45


Main Menu


====================================
====================================
```



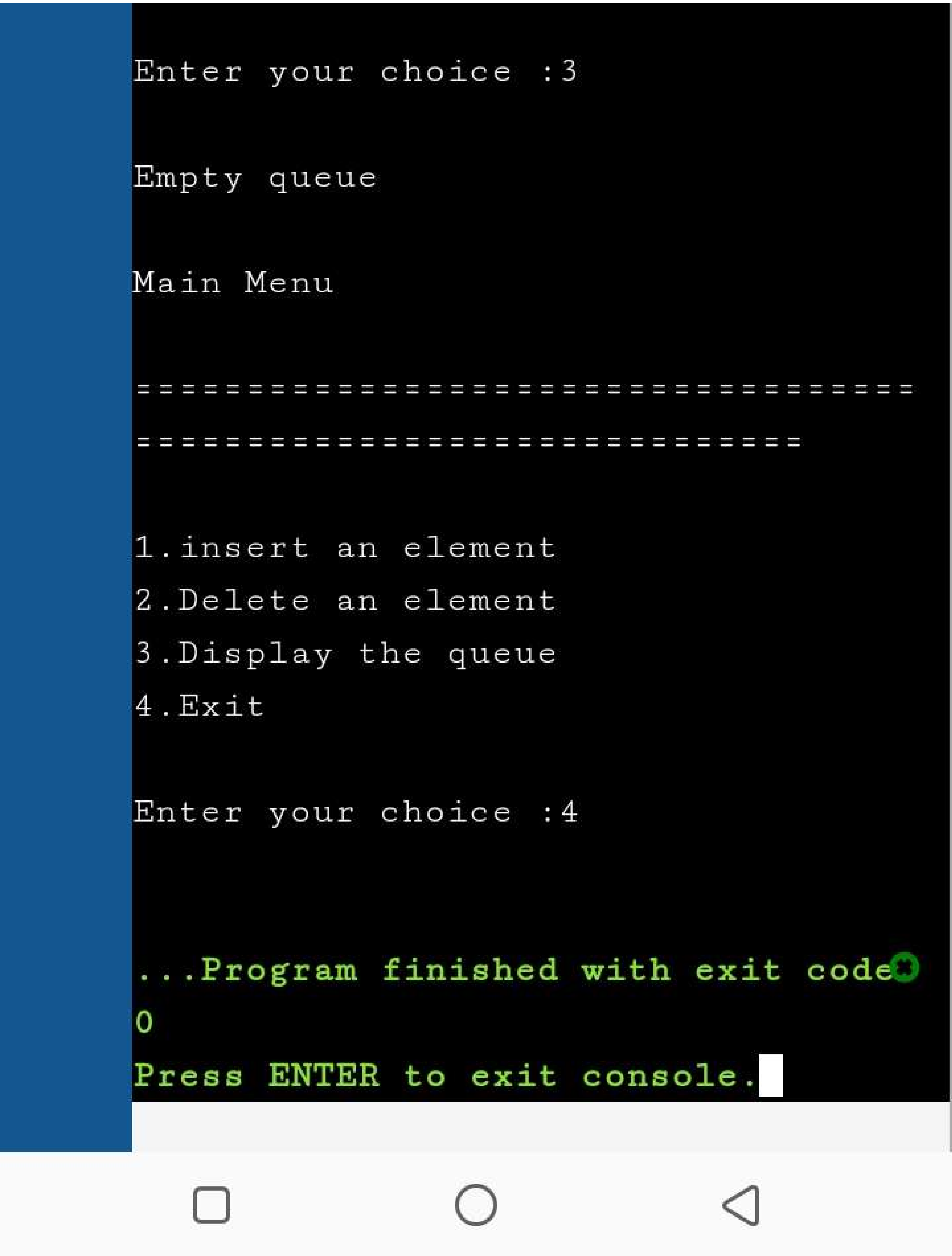
```
        ptr = front;
        front = front -> n
        free(ptr);
    }
```

```
Main Menu

===================================
==============================

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice :1

Enter value:
45

Main Menu

===================================
==============================

1.insert an element
2.Delete an element
3.Display the queue
```



```
            ptr = front;
            front = front -> n
            free(ptr);
        }
```

```
4.Exit

Enter your choice :2

Main Menu

===================================
==============================

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice :3

Empty queue

Main Menu

===================================
==============================

1.insert an element
```



```
Enter your choice :3

Empty queue

Main Menu

==================================
=============================

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice :4


...Program finished with exit code
0
Press ENTER to exit console.
```

Q. Program to implement queue using linked list

Program

```
#include <stdio.h>
#include <stdlib.h>
Struct node
{
 int data;
 struct node *next;
};
Struct node * front;
Struct node rear;
void insert();
void delete();
void display();
void main()
{
```

```
int choice;
while (choice != 4)
 {
  printf ("\n Main Menu");
  printf ("\n ============ \n");
Printf ("\n 1. insert an element \n 2. Delete an
  element \n 3. Display the queue \n 4.
  Exit \n");
printf ("\n Enter your choice: ");
Scanf ("%d", &choice);
 Switch (choice)
 {
 Case 1:
 insert();
 break;
 Case 2:
 delete();
 break.
Case 3:
 display();
 break;
 Case 4:
 exit(0);
 break;
 default:
 printf("\n Enter valid choice:\n");
  }
 }
}
void insert()
 {
 struct node * ptr;
 int item;
  ptr (struct node *)malloc (sizeof ( struct
        node));
   if (ptr==NULL)
```

```
{
 printf ("\n OVERFLOW\n");
return;
}
else
 {
 printf("\n Enter value!\n");
 scanf ("%d",&item);
 ptr->data =item;
 if (front == NULL)

 {   front=ptr;
     rear=ptr;
     front->next =NULL;
     rear-> next =NULL;
  }
 else
 {
   rear->next =ptr;
  rear=ptr;
  rear->next=NULL;
  }
 }
}

void delete()
{
  struct node *ptr;
   if(front==NULL)
   {
   printf("\n UNDERFLOW\n");
    return;
  }
 else
   ptr=front;
    front=front->next;
     free(ptr);
```

```
  }
 }
void diplay()
 {
 struct node *ptr;
  ptr=front;
  if(front==NULL)
   {
   printf("\n Empty queue\n");

   }
  else
   {
   printf("\n ~~Empty~~ display values...\n");
   while(ptr!=NULL)
     {
     printf("\n%d\n",ptr->data);
       ptr=ptr->next;
   }
 }
}
```

Output

```
Main menu
= = = = = = = = = = =
1. Insert an element
2. Delete an element
3. Display the queue
4. Exit
Enter your choice : 1
Enter value:
45
Main menu
= = = = = = = = = = =
1. insert an element
2. Delete an element
3. Display the queue
4. Exit
Enter your choice : 2
```

```
Main Menu

  = = = = = = = = = = =

1. insert an element
2. delete an element
3. Display the queue
4. exit
 Enter your choice:3

Empty queue

 Main Menu

  = = = = = = = = = = = =

1. insert an element
2. delete an element
3. display the queue
4. Exit

Enter your choice:4

 ... program finished with exit code 0
```