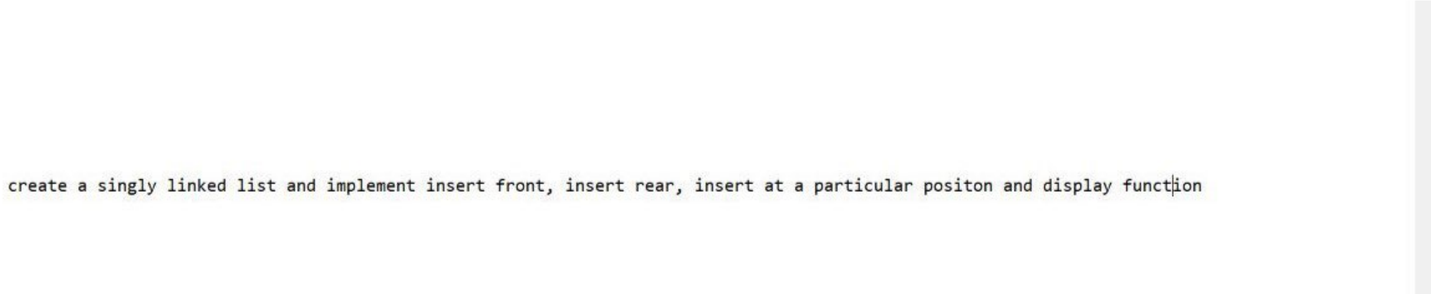
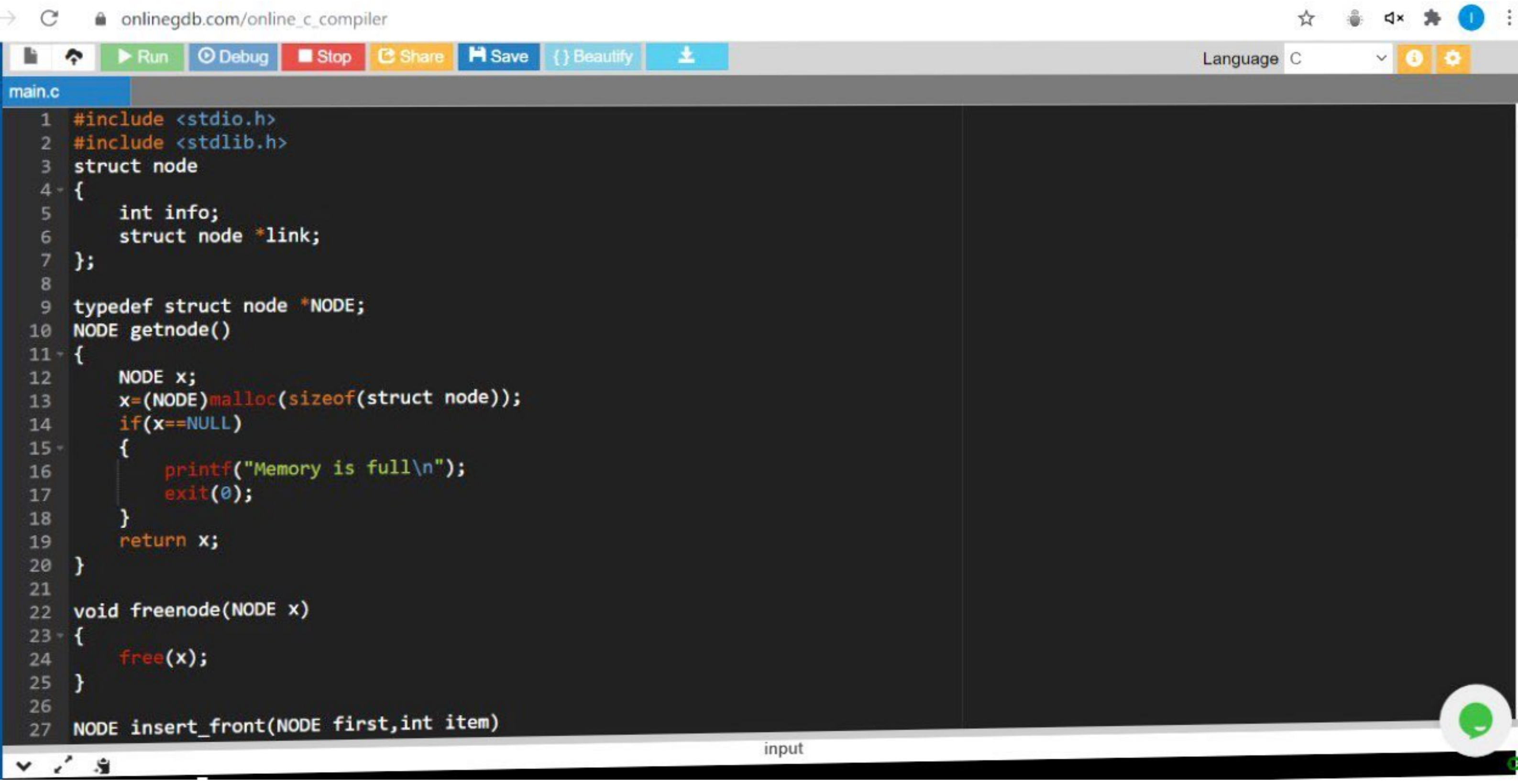
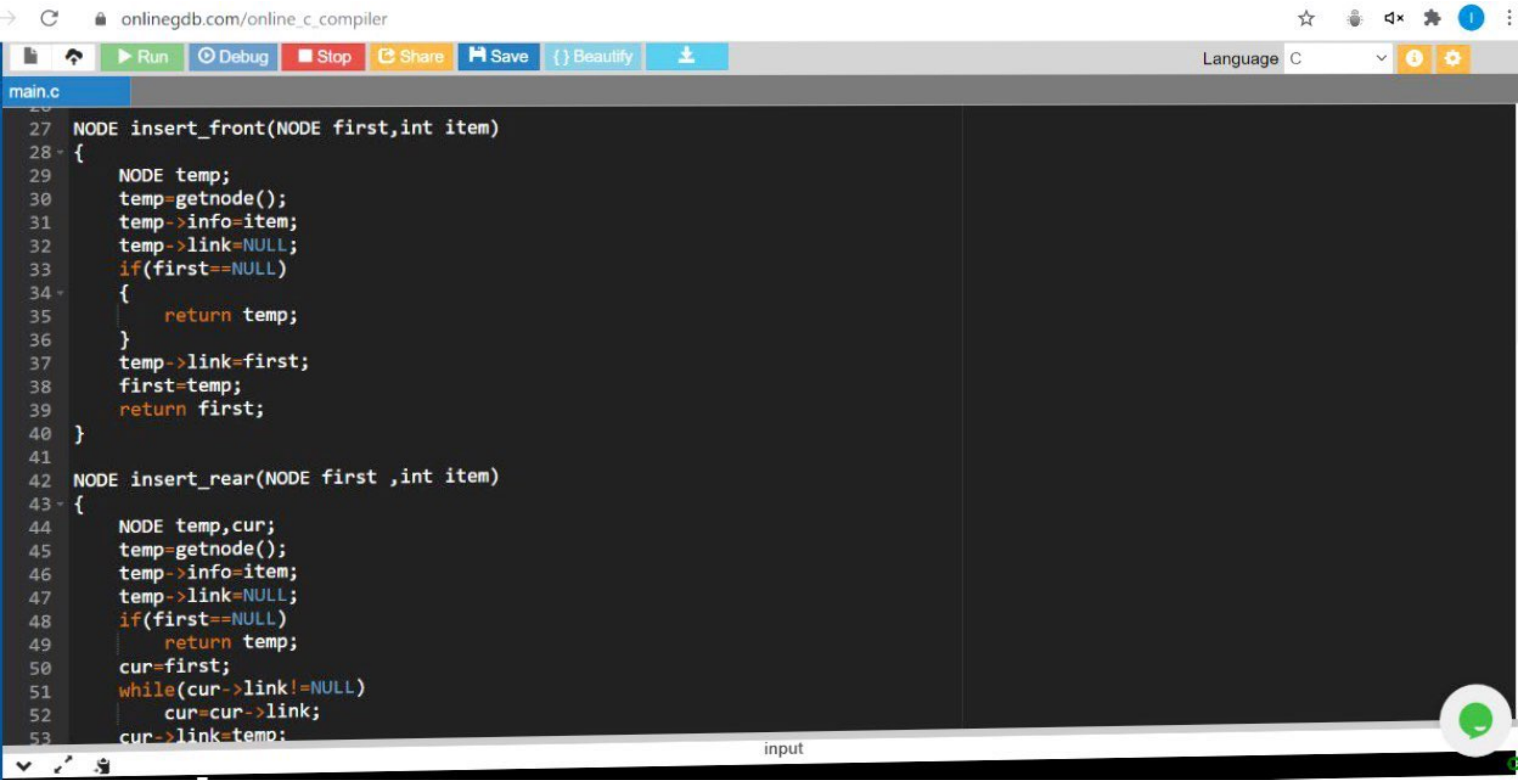


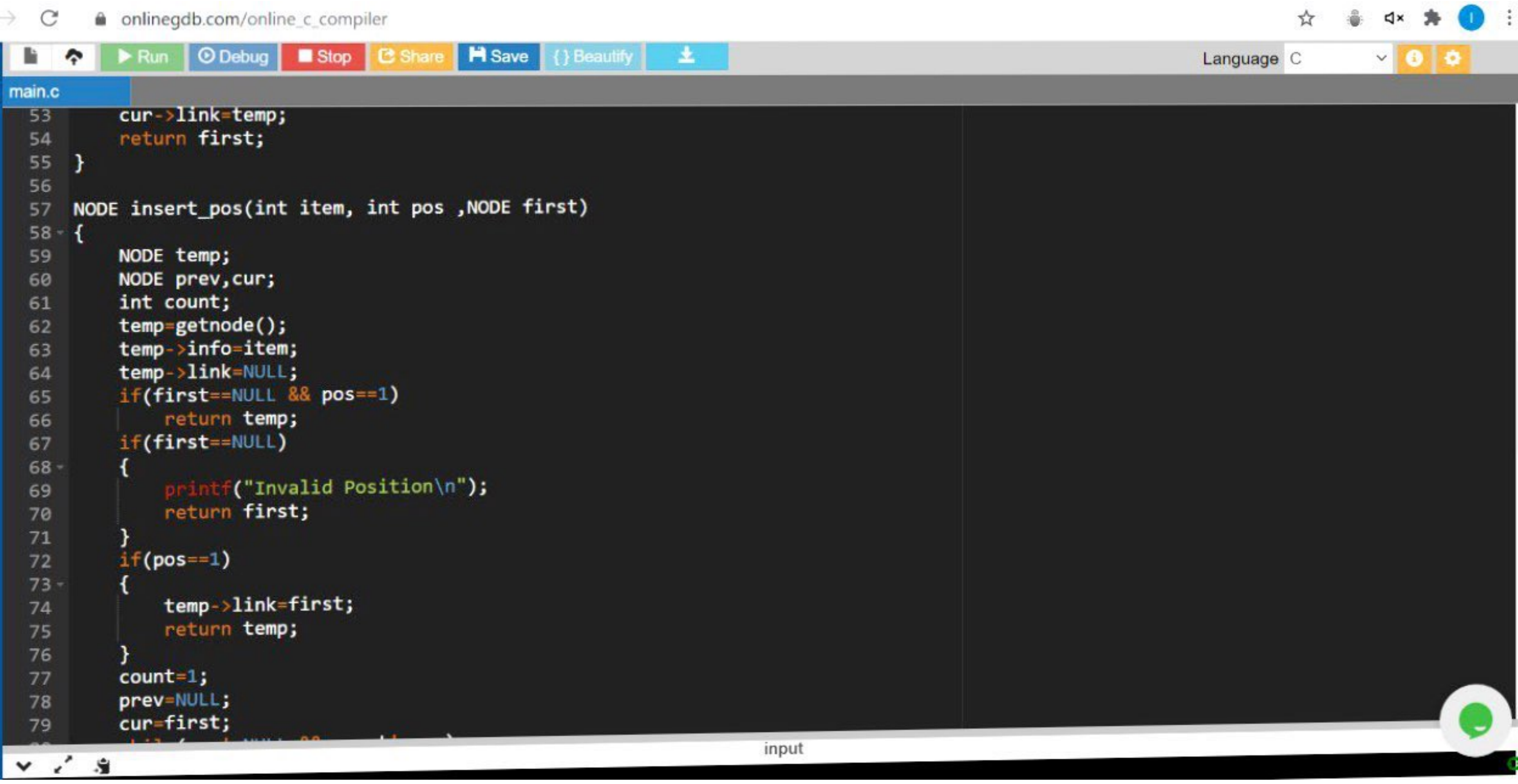
create a singly linked list and implement insert front, insert rear, insert at a particular positon and display function



```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int info;
    struct node *link;
};

typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if(x==NULL)
    {
        printf("Memory is full\n");
        exit(0);
    }
    return x;
}

void freenode(NODE x)
{
    free(x);
}

NODE insert_front(NODE first,int item)
```

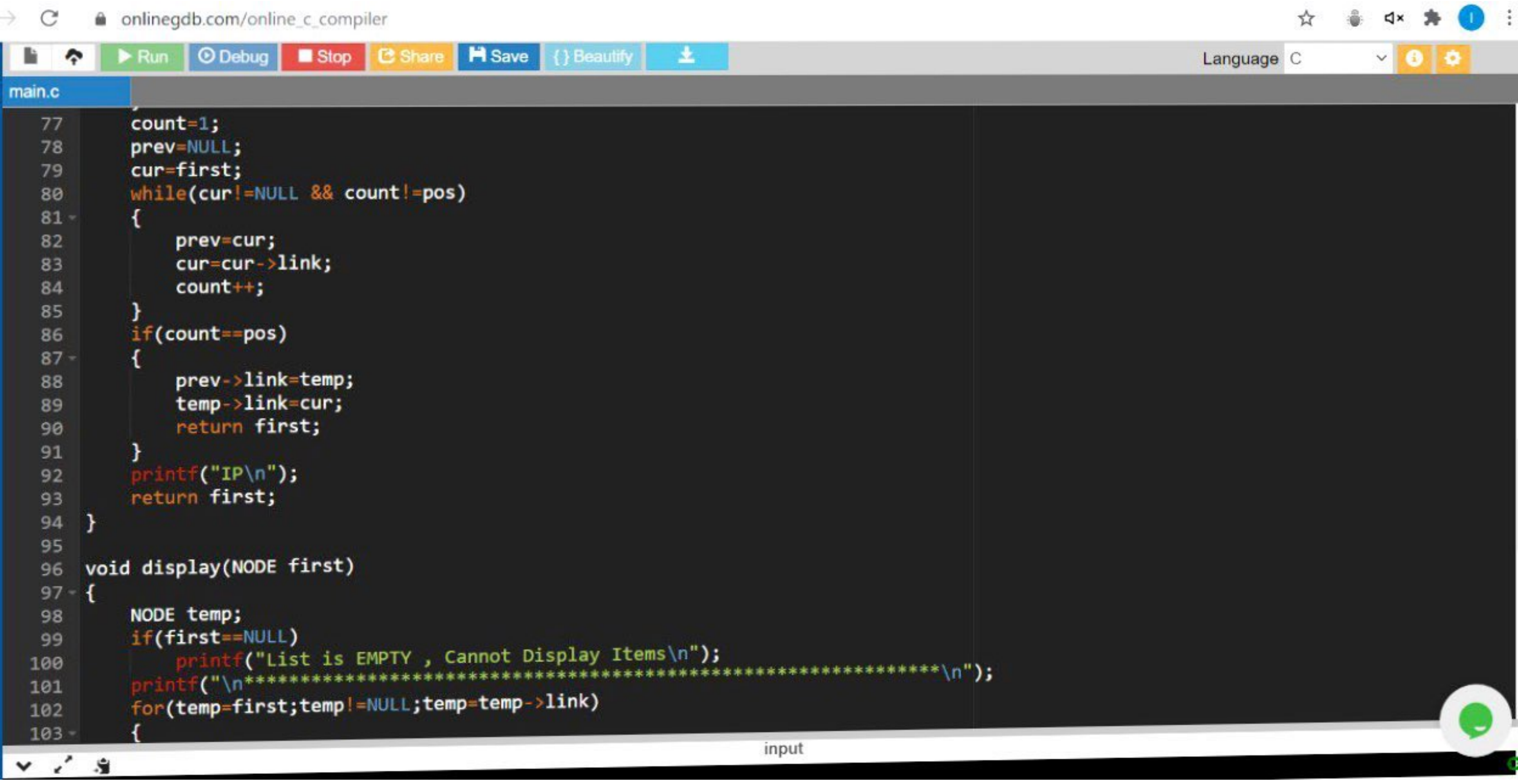


```
NODE insert_front(NODE first,int item)
{
    NODE temp;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL)
    {
        return temp;
    }
    temp->link=first;
    first=temp;
    return first;
}

NODE insert_rear(NODE first ,int item)
{
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL)
        return temp;
    cur=first;
    while(cur->link!=NULL)
        cur=cur->link;
    cur->link=temp;
```



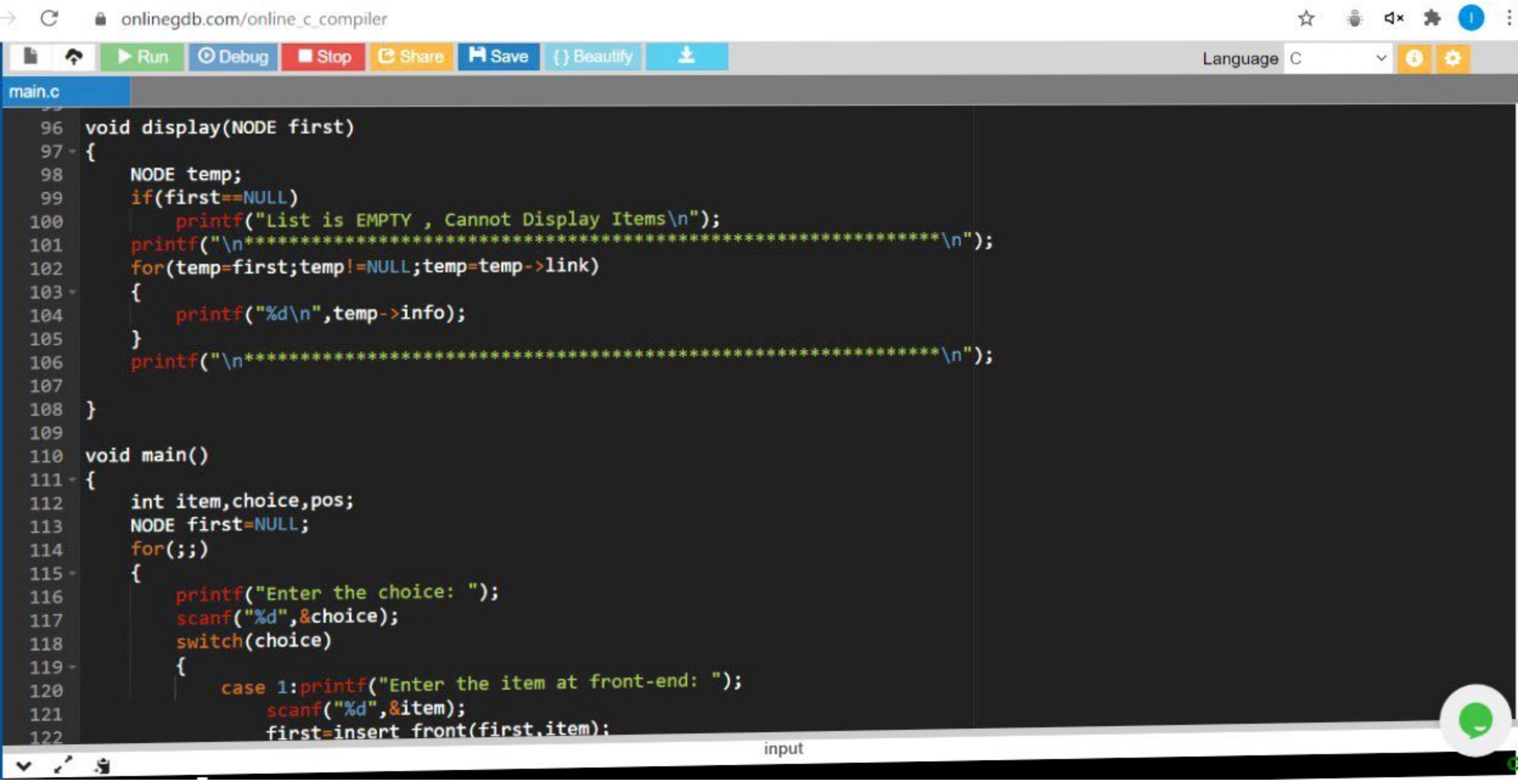
```c
    cur->link=temp;
    return first;
}

NODE insert_pos(int item, int pos ,NODE first)
{
    NODE temp;
    NODE prev,cur;
    int count;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if(first==NULL && pos==1)
        return temp;
    if(first==NULL)
    {
        printf("Invalid Position\n");
        return first;
    }
    if(pos==1)
    {
        temp->link=first;
        return temp;
    }
    count=1;
    prev=NULL;
    cur=first;
```

Scanned with CamScanner



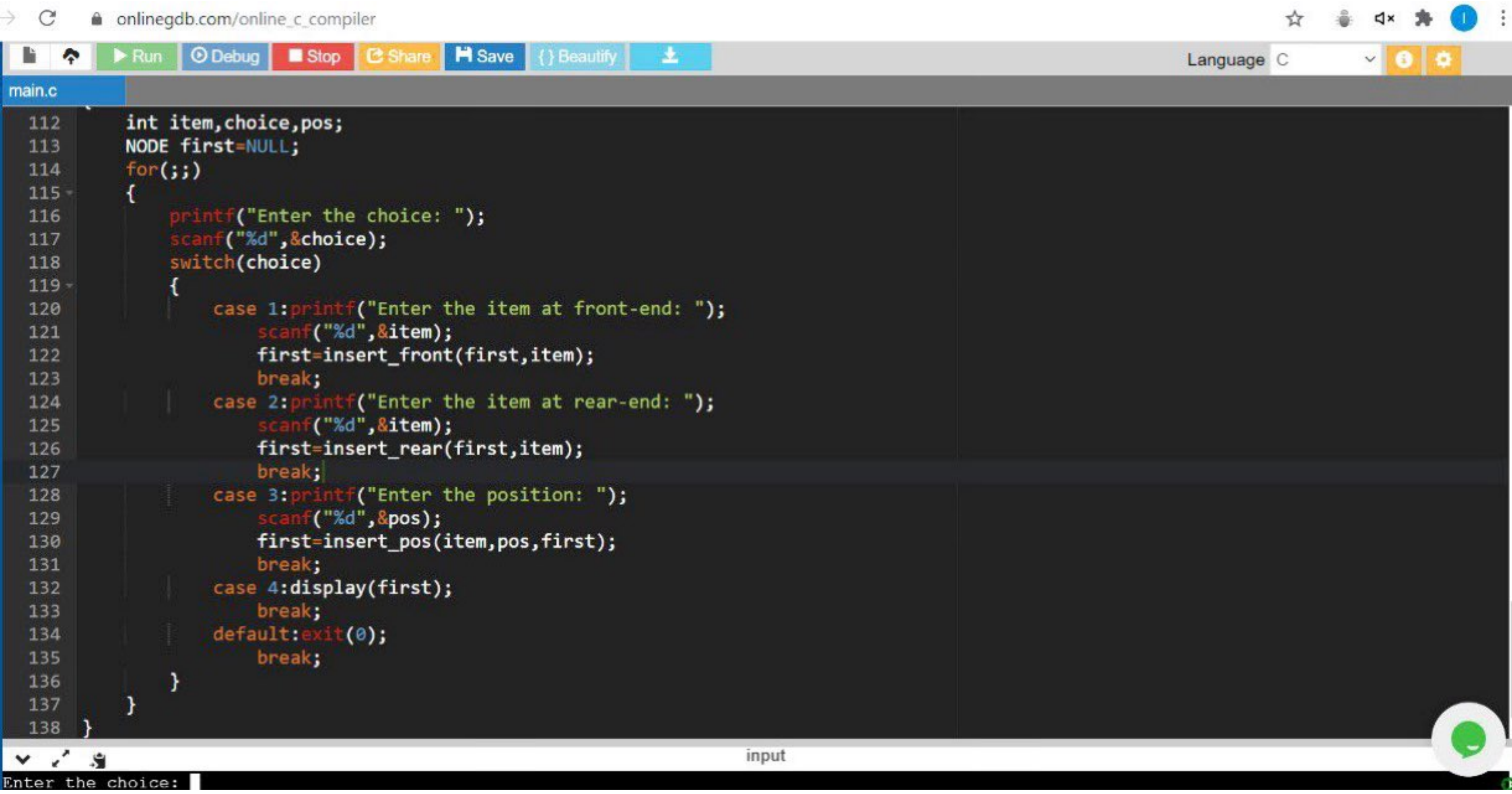
```c
    count=1;
    prev=NULL;
    cur=first;
    while(cur!=NULL && count!=pos)
    {
        prev=cur;
        cur=cur->link;
        count++;
    }
    if(count==pos)
    {
        prev->link=temp;
        temp->link=cur;
        return first;
    }
    printf("IP\n");
    return first;
}

void display(NODE first)
{
    NODE temp;
    if(first==NULL)
        printf("List is EMPTY , Cannot Display Items\n");
    printf("\n**************************************************************\n");
    for(temp=first;temp!=NULL;temp=temp->link)
    {
```

Scanned with CamScanner



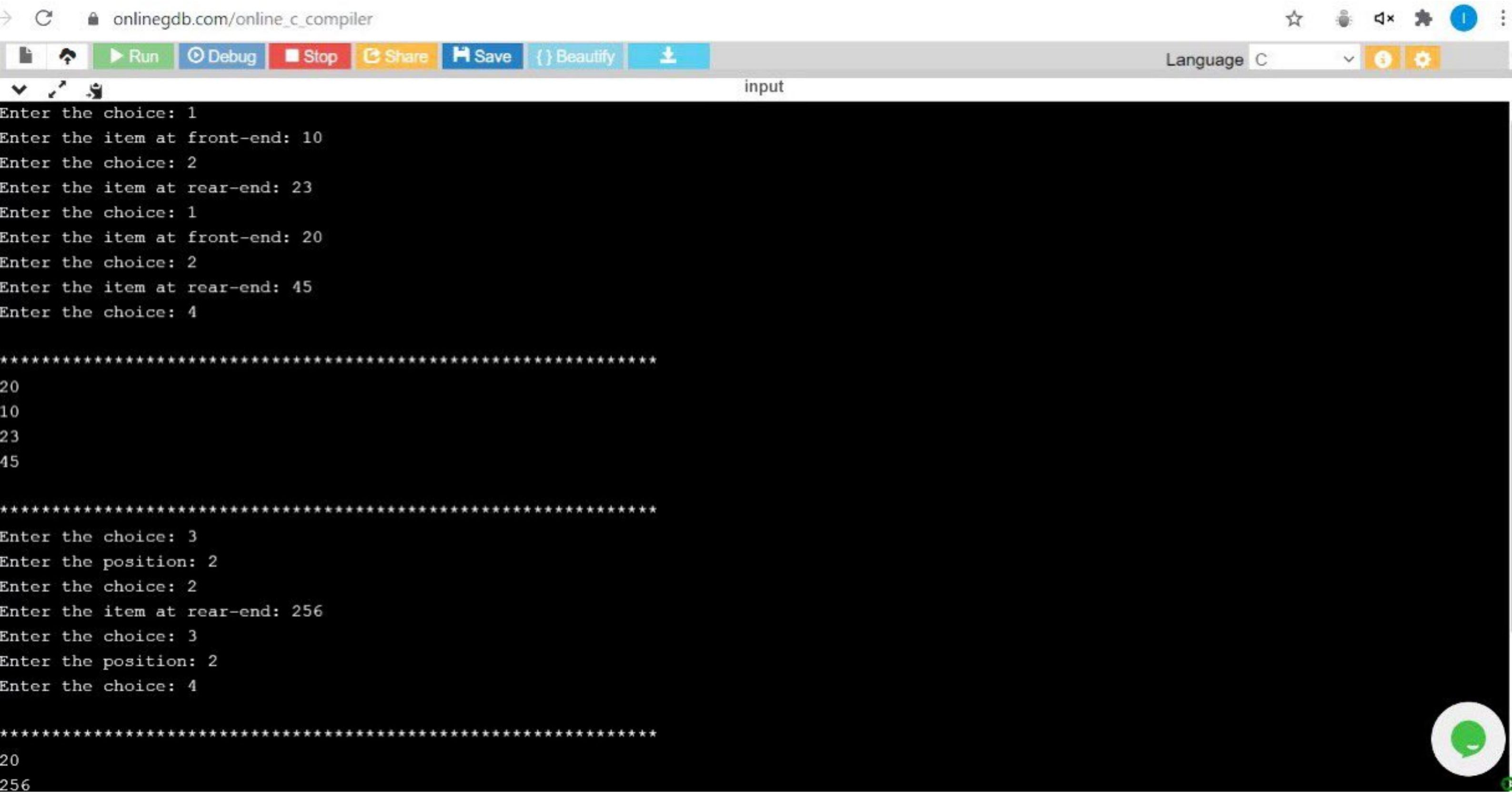
```c
void display(NODE first)
{
    NODE temp;
    if(first==NULL)
        printf("List is EMPTY , Cannot Display Items\n");
    printf("\n***************************************************************\n");
    for(temp=first;temp!=NULL;temp=temp->link)
    {
        printf("%d\n",temp->info);
    }
    printf("\n***************************************************************\n");

}

void main()
{
    int item,choice,pos;
    NODE first=NULL;
    for(;;)
    {
        printf("Enter the choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the item at front-end: ");
                scanf("%d",&item);
                first=insert_front(first,item);
```

Scanned with CamScanner

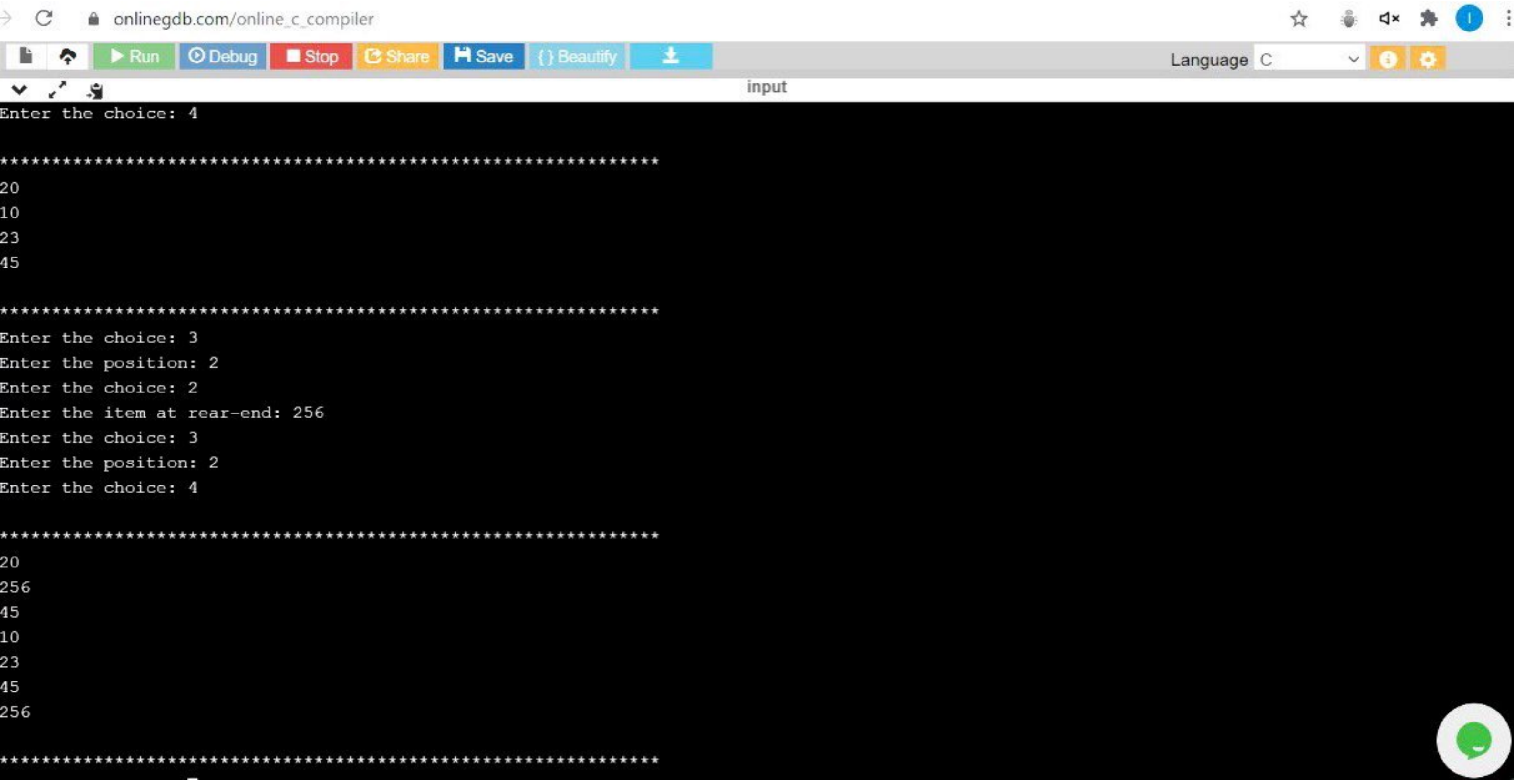


```c
    int item,choice,pos;
    NODE first=NULL;
    for(;;)
    {
        printf("Enter the choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the item at front-end: ");
                scanf("%d",&item);
                first=insert_front(first,item);
                break;
            case 2:printf("Enter the item at rear-end: ");
                scanf("%d",&item);
                first=insert_rear(first,item);
                break;
            case 3:printf("Enter the position: ");
                scanf("%d",&pos);
                first=insert_pos(item,pos,first);
                break;
            case 4:display(first);
                break;
            default:exit(0);
                break;
        }
    }
}
```

```
Enter the choice: 
```



```
Enter the choice: 1
Enter the item at front-end: 10
Enter the choice: 2
Enter the item at rear-end: 23
Enter the choice: 1
Enter the item at front-end: 20
Enter the choice: 2
Enter the item at rear-end: 45
Enter the choice: 4

***************************************************************
20
10
23
45

***************************************************************
Enter the choice: 3
Enter the position: 2
Enter the choice: 2
Enter the item at rear-end: 256
Enter the choice: 3
Enter the position: 2
Enter the choice: 4

***************************************************************
20
256
```



```
Enter the choice: 4

****************************************************************
20
10
23
45

****************************************************************
Enter the choice: 3
Enter the position: 2
Enter the choice: 2
Enter the item at rear-end: 256
Enter the choice: 3
Enter the position: 2
Enter the choice: 4

****************************************************************
20
256
45
10
23
45
256

****************************************************************
```