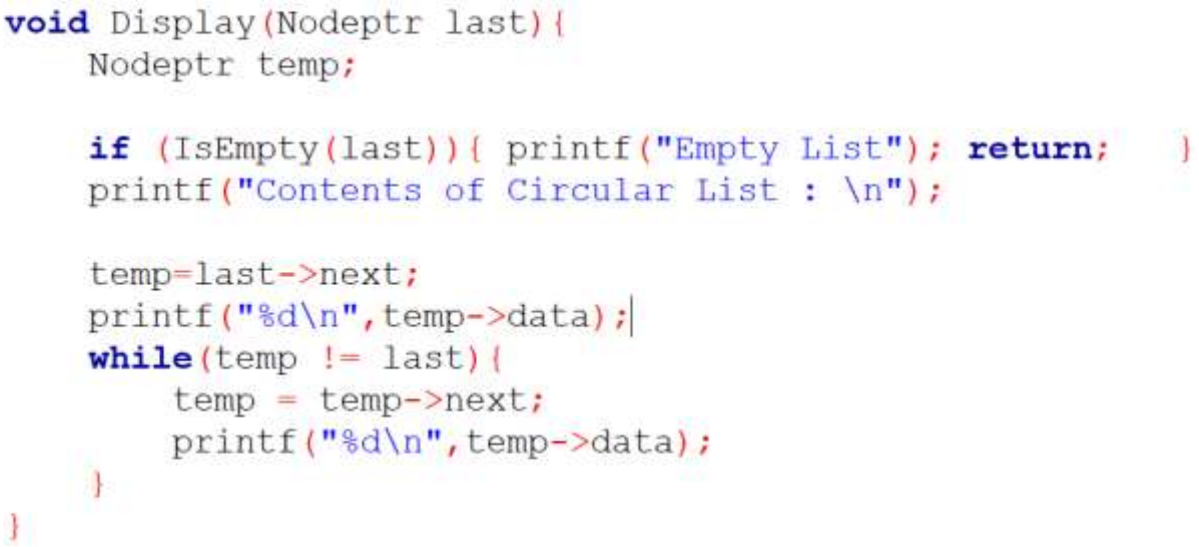
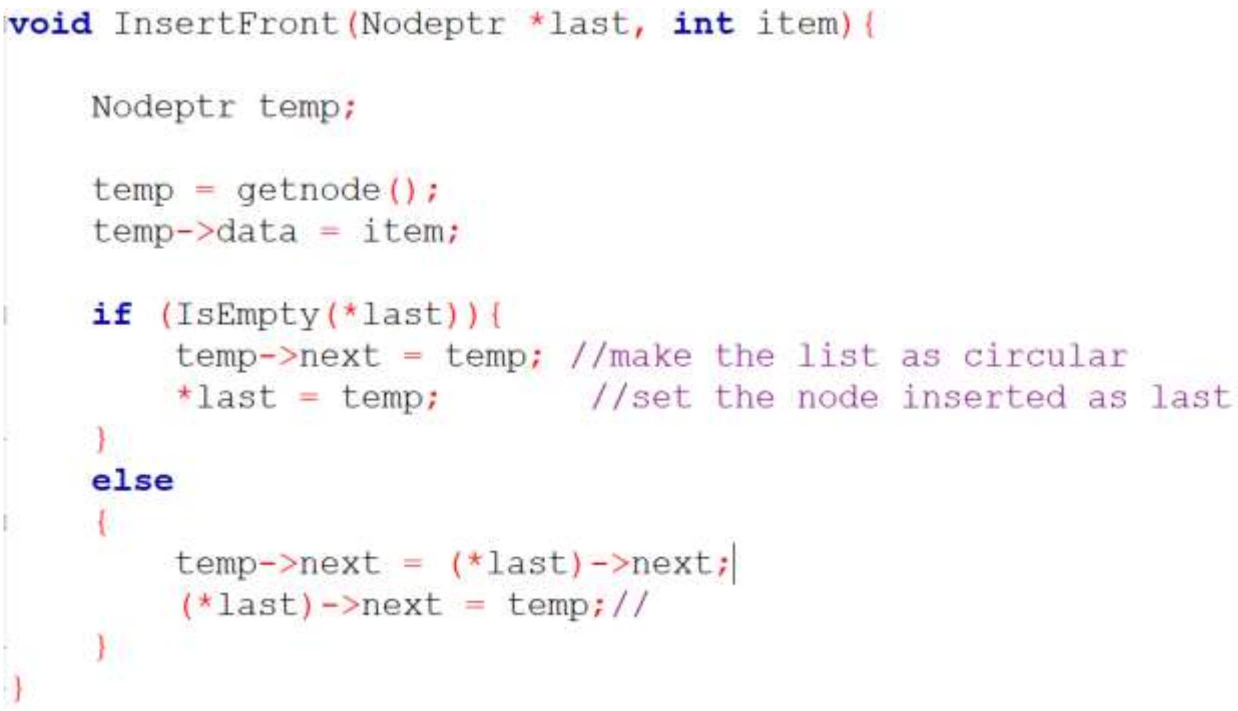
Singly linked Circular List

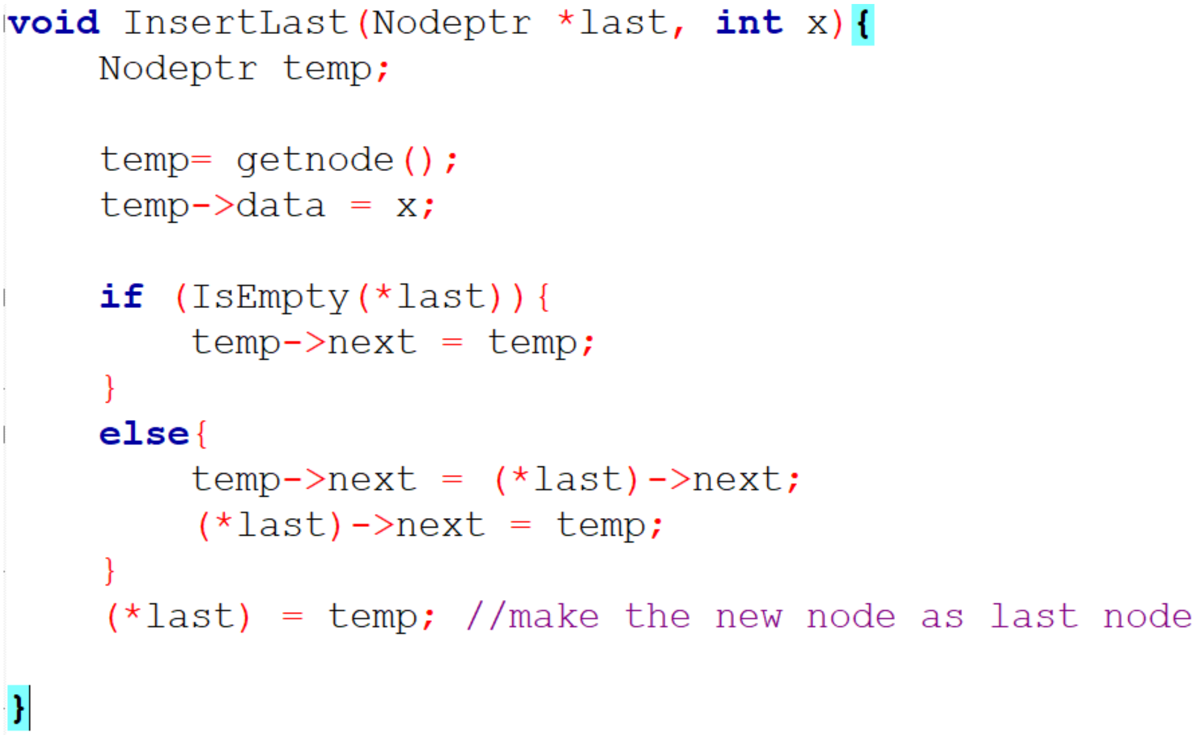
```c
struct node{
    int data;
    Nodeptr next;
};
int IsEmpty(Nodeptr last){
    return (last == NULL ? 1:0);
}
Nodeptr getnode(){
    Nodeptr temp;

    temp = (Nodeptr) malloc(sizeof(struct node));

    return temp;
}
```



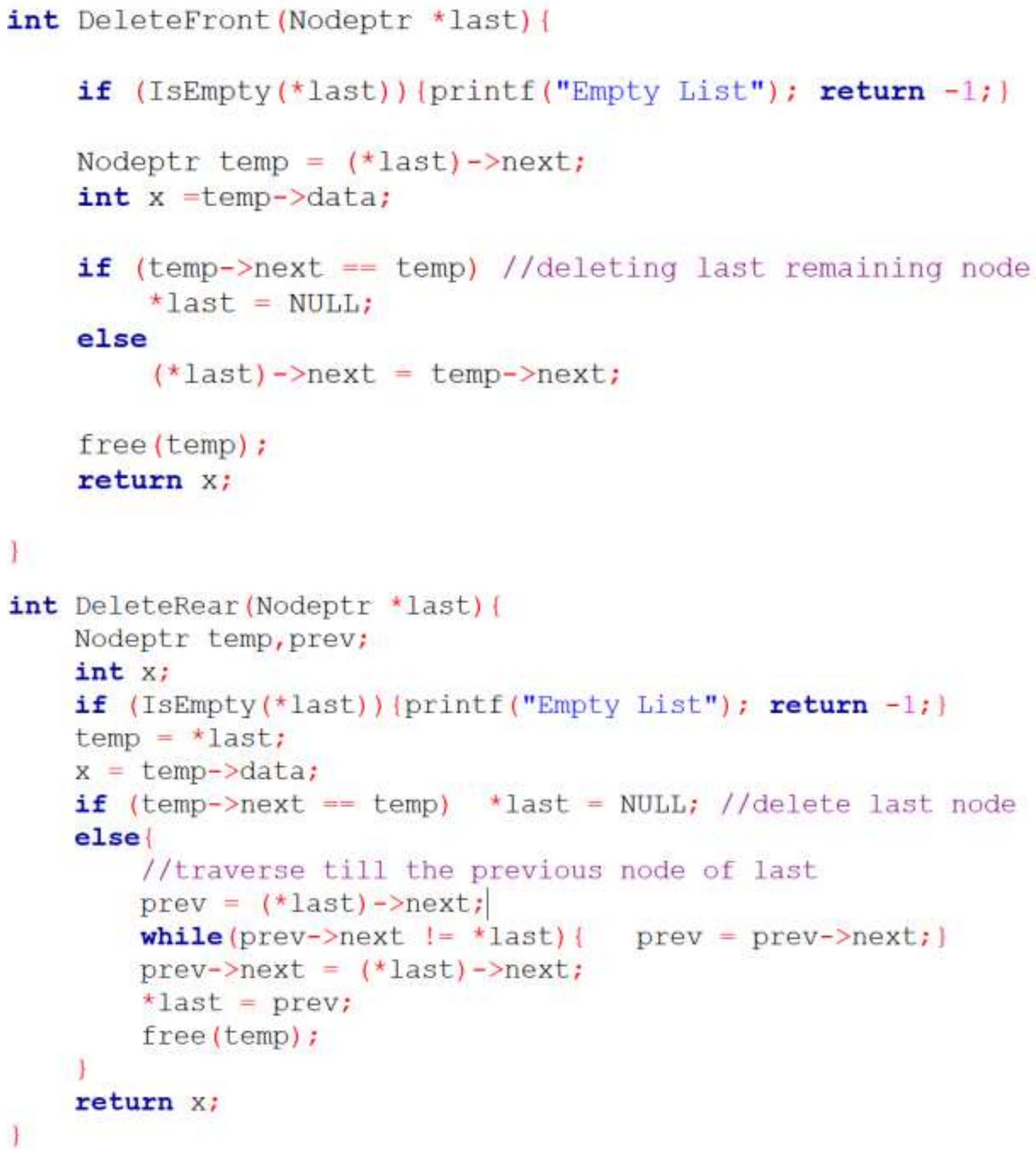
```c
void Display(Nodeptr last){
    Nodeptr temp;

    if (IsEmpty(last)){ printf("Empty List"); return;    }
    printf("Contents of Circular List : \n");

    temp=last->next;
    printf("%d\n",temp->data);
    while(temp != last){
        temp = temp->next;
        printf("%d\n",temp->data);
    }
}
```



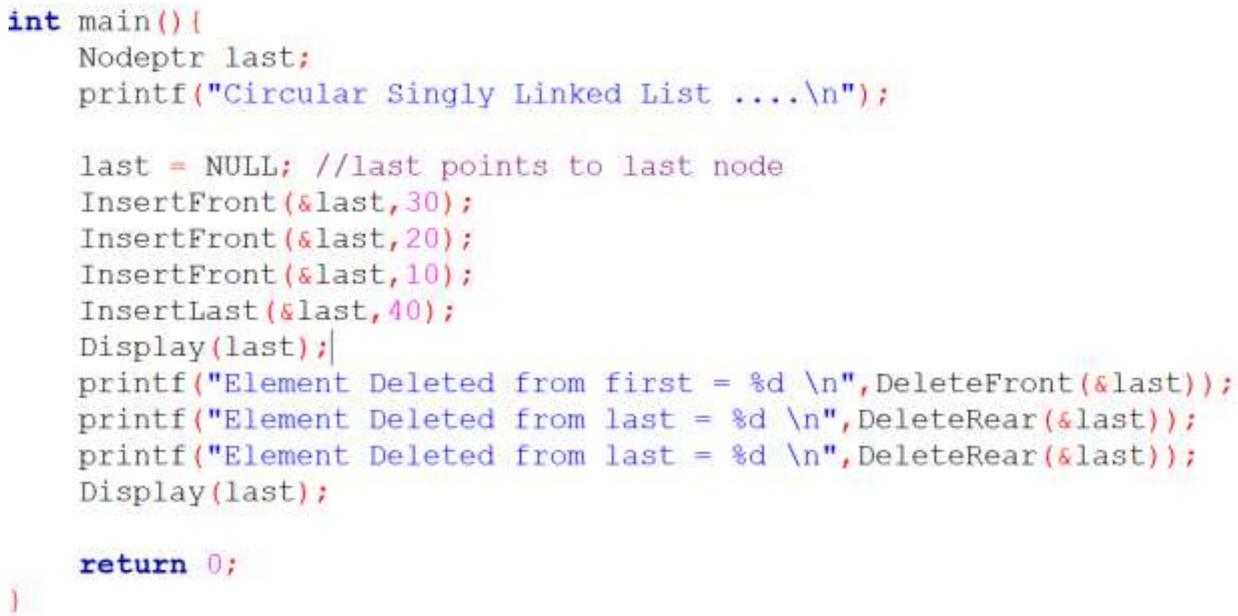
```
void InsertFront(Nodeptr *last, int item){

    Nodeptr temp;

    temp = getnode();
    temp->data = item;

    if (IsEmpty(*last)){
        temp->next = temp; //make the list as circular
        *last = temp;       //set the node inserted as last
    }
    else
    {
        temp->next = (*last)->next;
        (*last)->next = temp;//
    }
}
```



```
void InsertLast(Nodeptr *last, int x){
    Nodeptr temp;

    temp= getnode();
    temp->data = x;

    if (IsEmpty(*last)){
        temp->next = temp;
    }
    else{
        temp->next = (*last)->next;
        (*last)->next = temp;
    }
    (*last) = temp; //make the new node as last node

}
```



```
int DeleteFront(Nodeptr *last){

    if (IsEmpty(*last)){printf("Empty List"); return -1;}

    Nodeptr temp = (*last)->next;
    int x =temp->data;

    if (temp->next == temp) //deleting last remaining node
        *last = NULL;
    else
        (*last)->next = temp->next;

    free(temp);
    return x;

}

int DeleteRear(Nodeptr *last){
    Nodeptr temp,prev;
    int x;
    if (IsEmpty(*last)){printf("Empty List"); return -1;}
    temp = *last;
    x = temp->data;
    if (temp->next == temp)  *last = NULL; //delete last node
    else{
        //traverse till the previous node of last
        prev = (*last)->next;
        while(prev->next != *last){   prev = prev->next;}
        prev->next = (*last)->next;
        *last = prev;
        free(temp);
    }
    return x;
}
```



```c
int main(){
    Nodeptr last;
    printf("Circular Singly Linked List ....\n");

    last = NULL; //last points to last node
    InsertFront(&last,30);
    InsertFront(&last,20);
    InsertFront(&last,10);
    InsertLast(&last,40);
    Display(last);
    printf("Element Deleted from first = %d \n",DeleteFront(&last));
    printf("Element Deleted from last = %d \n",DeleteRear(&last));
    printf("Element Deleted from last = %d \n",DeleteRear(&last));
    Display(last);

    return 0;
}
```