e ✕ linkedlistinsertion.c ✕ circularqueue.c ✕

```c
#include<stdio.h>
#include<stdlib.h>
struct Node{
int data;
struct Node *next;
};
struct Node *head=NULL;
void createList(int n){
struct Node *newNode,*temp;
int data,i;
if(n<=0){printf("Number of nodes should be greater than 0\n");
return ;}
for(i=1;i<=n;i++)
{
    newNode=(struct Node *)malloc(sizeof(struct Node));
    if(newNode==NULL){
        printf("Memory allocation failed\n");
        return;
    }
    printf("Enter data for node %d: ",i);
    scanf("%d",&data);
    newNode ->data=data;
    newNode->next=NULL;
    if(head==NULL){
        head=newNode;
    }
    else{
        temp->next=newNode;
    }
    temp=newNode;
    }
    printf("\nLinked list created successfully\n");
}
void insertAtBeginning(int data){
struct Node *newNode=(struct Node*)malloc (sizeof(struct Node));
newNode->data=data;
newNode->next=head;
```

```
newNode->next=head;
head=newNode;
printf("Node inserted at the beginning\n");
}
void insertionAtEnd(int data){
struct Node *newNode=(struct Node*)malloc (sizeof(struct Node));
newNode->data=data;
newNode->next=NULL;
if(head==NULL){
    head=newNode;
}
else{
    struct Node *temp=head;
    while(temp->next!=NULL)
        temp=temp->next;
    temp->next=newNode;
}
printf("Node inserted at the end\n");
}
void insertAtPosition(int data,int pos){
int i;
struct Node *newNode,*temp=head;
if(pos<1){
    printf("Invalid position\n");
    return;
}
if(pos==1){
    insertAtBeginning(data);
    return;
}
newNode=(struct Node*)malloc(sizeof(struct Node));
newNode->data=data;
for(i=1;i<pos-1 && temp!=NULL;i++)
    temp=temp->next;
if(temp==NULL){
    printf("Position out of range\n");
    free(newNode);
```

```
    free(newNode);
}else{
newNode->next=temp->next;
temp->next=newNode;
printf("Node inserted at position %d\n",pos);
}

}
void displayList(){
struct Node *temp=head;
if(head==NULL){
    printf("List is empty\n");
}
printf("\nLinked List: ");
while (temp!=NULL){
    printf("%d -> ",temp->data);
    temp=temp->next;
}
printf("NULL\n");
}
int main(){
int choice,n,data,pos;
while(1){
    printf("\n---SIngly Linked List Operations---\n");
    printf("1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit\n");
    printf("Enter your choice: ");
    scanf("%d",&choice);
    switch(choice){
    case 1:printf("Enter num of nodes");
           scanf("%d" ,&n);
           createList(n);
           break;
    case 2:printf("Enter data to insert :");
            scanf("%d",&data);
            insertAtBeginning(data);
            break;
    case 3:printf("enter data and position: ");
```

```c
int main(){
 int choice,n,data,pos;
while(1){
    printf("\n---SIngly Linked List Operations---\n");
    printf("1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit\n");
    printf("Enter your choice: ");
    scanf("%d",&choice);
    switch(choice){
    case 1:printf("Enter num of nodes");
           scanf("%d" ,&n);
           createList(n);
           break;
    case 2:printf("Enter data to insert :");
            scanf("%d",&data);
            insertAtBeginning(data);
            break;
    case 3:printf("enter data and position: ");
             scanf("%d %d",&data,&pos);
             insertAtPosition(data,pos);
             break;
    case 4:
        printf("Enter data to insert:");
        scanf("%d",&data);
        insertionAtEnd(data);break;
    case 5:
        displayList();
        break;
    case 6:printf("exiting...");
           exit(0);

    default:printf("Invalid choice Try again\n");
    }
}
return 0;
}
```

```
---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 1
Enter num of nodes5
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30
Enter data for node 4: 40
Enter data for node 5: 50

Linked list created successfully

---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 2
Enter data to insert :60
Node inserted at the beginning

---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 3
enter data and position: 35
4
Node inserted at position 4

---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 4
Enter data to insert:55
Node inserted at the end

---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 5

Linked List: 60 -> 10 -> 20 -> 35 -> 30 -> 40 -> 50 -> 55 -> NULL

---SIngly Linked List Operations---
1.Create Linked List 2.Insert at Beginning 3.Insert at any position 4.insert at end 5.display 6.exit
Enter your choice: 6
exiting...
Process returned 0 (0x0)   execution time : 88.720 s
Press any key to continue.
```