

## PROGRAM- 4 AND 5

The screenshot shows the Code::Blocks 20.03 IDE interface. The main window displays C code for inserting items at the beginning of a linked list. The code includes functions for inserting at the beginning and inserting after a specific position. The interface includes a toolbar, menu bar, and status bar indicating the file is p3.c, line 240, col 20, pos 5393.

```
34 struct node* insertslbeg(struct node* start,int item)
35 {
36     struct node* p;
37     p=(struct node*)malloc(sizeof(struct node));
38     p->info=item;
39     if(start==NULL)
40     {
41         p->next=NULL;
42         start=p;
43     }
44     else
45     {
46         p->next=start;
47         start=p;
48     }
49     return start;
50 }
51 struct node* insertslend(struct node* start,int item)
52 {
53     struct node* p,*temp;
54     p=(struct node*)malloc(sizeof(struct node));
55     p->info=item;
56     if(start==NULL)
57     {
58         p->next=NULL;
59         start=p;
60     }
61     else
62     {
63         temp=start;
64         while(temp->next!=NULL)
65         {
66             temp=temp->next;
67         }
68         temp->next=p;
69         p->next=NULL;
70     }
71     return start;
72 }
73 struct node* insertslpos(struct node* start,int item,int pos)
74 {
75     struct node *p,*temp;
76     int i;
77     p = (struct node*)malloc(sizeof(struct node));
78     if (start == NULL) {
79         printf("Memory allocation failed\n");
80         return start;
81     }
82     p->info = item;
83     if (pos == 1) {
84         p->next = start;
85         start = p;
86         return start;
87     }
88     temp = start;
89     for (i = 1; i < pos - 1 && temp != NULL; i++)
90     {
91         temp = temp->next;
92     }
93     if (temp == NULL) {
94         printf("Position out of range!\n");
95         free(p);
96         return start;
97     }
98 }
```

The screenshot shows the Code::Blocks 20.03 IDE interface. The main window displays C code for inserting items at the beginning of a linked list. The code includes functions for inserting at the beginning and inserting after a specific position. The interface includes a toolbar, menu bar, and status bar indicating the file is p3.c, line 240, col 20, pos 5393.

```
67 }
68     temp->next=p;
69     p->next=NULL;
70 }
71 return start;
72 }
73 struct node* insertslpos(struct node* start,int item,int pos)
74 {
75     struct node *p,*temp;
76     int i;
77     p = (struct node*)malloc(sizeof(struct node));
78     if (start == NULL) {
79         printf("Memory allocation failed\n");
80         return start;
81     }
82     p->info = item;
83     if (pos == 1) {
84         p->next = start;
85         start = p;
86         return start;
87     }
88     temp = start;
89     for (i = 1; i < pos - 1 && temp != NULL; i++)
90     {
91         temp = temp->next;
92     }
93     if (temp == NULL) {
94         printf("Position out of range!\n");
95         free(p);
96         return start;
97     }
98 }
```

\*p3.c - Code::Blocks 20.03

```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> insertsllpos(struct node* start, int item, int pos) : node
Management Start here X p3.exe X *p3.c X
Projects Files FSy *
Workspace
1 Projects 2 Files 3 FSy
97
98     free(p);
99     return start;
100 }
101 p->next = temp->next;
102 temp->next = p;
103 return start;
104 }
105 struct node* deletesllbeg(struct node* start)
106 {
107     struct node* temp;
108     if(start==NULL)
109     {
110         printf("Linked list is empty");
111         return start;
112     }
113     else
114     {
115         temp=start;
116         printf("Deleted element from beginning is:%d",temp->info);
117         start=start->next;
118         free(temp);
119     }
120     return start;
121 }
122
123
124 struct node* deletesllend(struct node* start)
125 {
126     struct node* temp,*follow;
127     if(start==NULL)
128     {
129         printf("Linked list is empty");
130     }
131     else if(start->next==NULL)
132     {
133         temp=start;
134         printf("Deleted element from end is:%d",temp->info);
135         free(temp);
136         start=NULL;
137         return start;
138     }
139     else{
140         temp=start;
141         while(temp->next!=NULL)
142         {
143             follow=temp;
144             temp=temp->next;
145             follow->next=NULL;
146             printf("Deleted element from end is:%d",temp->info);
147             free(temp);
148         }
149         return start;
150     }
151 }
152
153 struct node* deletesllpos(struct node* start, int pos)
154 {
155     struct node* temp,*follow;
156     int i;
157
158     if(start == NULL)
159     {
160         printf("Linked list is empty\n");
161         return start;
162     }
163 }
```

C:\Users\gupta\Desktop\ritu dsal\p3.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 101, Col 20, Pos 2070 Insert Modified Read/Write default 09:37 ENG IN 22-11-2025

\*p3.c - Code::Blocks 20.03

```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> insertsllpos(struct node* start, int item, int pos) : node
Management Start here X p3.exe X *p3.c X
Projects Files FSy *
Workspace
130
131     return start;
132 }
133 else if(start->next==NULL)
134 {
135     temp=start;
136     printf("Deleted element from end is:%d",temp->info);
137     free(temp);
138     start=NULL;
139     return start;
140 }
141 else{
142     temp=start;
143     while(temp->next!=NULL)
144     {
145         follow=temp;
146         temp=temp->next;
147         follow->next=NULL;
148         printf("Deleted element from end is:%d",temp->info);
149         free(temp);
150     }
151     return start;
152 }
153
154 struct node* deletesllpos(struct node* start, int pos)
155 {
156     struct node* temp,*follow;
157     int i;
158
159     if(start == NULL)
160     {
161         printf("Linked list is empty\n");
162         return start;
163     }
164 }
```

C:\Users\gupta\Desktop\ritu dsal\p3.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 101, Col 20, Pos 2070 Insert Modified Read/Write default 09:37 ENG IN 22-11-2025

C:\Users\gupta\Desktop\ritu dsal\p3.c

```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> insertsllpos(struct node* start, int item, int pos) : node
Management Projects Files FSy Start here x p3.exe x *p3.c x
160     return start;
161 }
162
163 temp = start;
164
165 if (pos == 1) {
166     start = start->next;
167     printf("Deleted element is %d\n", temp->info);
168     free(temp);
169     return start;
170 }
171
172 follow = start;
173 for (i = 1; i < pos && temp != NULL; i++) {
174     follow = temp;
175     temp = temp->next;
176 }
177
178 if (temp == NULL) {
179     printf("Position out of range or invalid!\n");
180     return start;
181 }
182
183 follow->next = temp->next;
184 printf("Deleted element is %d\n", temp->info);
185 free(temp);
186 return start;
187 }
188
189 void display(struct node *start) {
190     struct node *temp = start;
191     printf("\nLinked List:");
192     while (temp != NULL)
193         temp = temp->next;
194     printf("NULL\n");
195 }
196
197
198
199
200 int main() {
201     struct node *start=NULL;
202     int choice,item,pos;
203     printf("Creation of linked list:\n");
204     start=createSLL();
205     display(start);
206     while(1)
207     {
208         printf("\n\n*****MENU*****\n\n");
209         printf("1.Insert at the beginning.\n2.Insert at the end.\n3.Insert at a given position.\n4.Delete at the beginning.\n5.Delete at the end.\n6.Delete at the given position.\n7.Exit\n");
210         scanf("%d",&choice);
211         switch(choice)
212         {
213             case 1:printf("Enter item to insert at beginning:");
214             scanf("%d",&item);
215             start=insertsllbeg(start,item);
216             display(start);
217             break;
218
219             case 2:printf("Enter item to insert at end:");
220             scanf("%d",&item);
221             start=insertsllend(start,item);
222             display(start);
223         }
224     }
225 }
```

C:\Users\gupta\Desktop\ritu dsal\p3.c

18°C Mostly clear

Search

Windows (CR+LF) WINDOWS-1252 Line 101, Col 20, Pos 2070 Insert Modified Read/Write default

ENG IN 09:37 22-11-2025

C:\Users\gupta\Desktop\ritu dsal\p3.c

```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> insertsllpos(struct node* start, int item, int pos) : node
Management Projects Files FSy Start here x p3.exe x *p3.c x
190     struct node *temp = start;
191     printf("\nLinked List:");
192     while (temp!=NULL) {
193         printf("%d ->", temp->info);
194         temp = temp->next;
195     }
196     printf("NULL\n");
197
198
199
200 int main() {
201     struct node *start=NULL;
202     int choice,item,pos;
203     printf("Creation of linked list:\n");
204     start=createSLL();
205     display(start);
206     while(1)
207     {
208         printf("\n\n*****MENU*****\n\n");
209         printf("1.Insert at the beginning.\n2.Insert at the end.\n3.Insert at a given position.\n4.Delete at the beginning.\n5.Delete at the end.\n6.Delete at the given position.\n7.Exit\n");
210         scanf("%d",&choice);
211         switch(choice)
212         {
213             case 1:printf("Enter item to insert at beginning:");
214             scanf("%d",&item);
215             start=insertsllbeg(start,item);
216             display(start);
217             break;
218
219             case 2:printf("Enter item to insert at end:");
220             scanf("%d",&item);
221             start=insertsllend(start,item);
222             display(start);
223         }
224     }
225 }
```

C:\Users\gupta\Desktop\ritu dsal\p3.c

18°C Mostly clear

Search

Windows (CR+LF) WINDOWS-1252 Line 101, Col 20, Pos 2070 Insert Modified Read/Write default

ENG IN 09:38 22-11-2025

The screenshot shows the Code::Blocks 20.03 IDE interface. The main window displays a C program named p3.c. The code implements a linked list with nodes containing integer values. It includes functions for inserting a new node at a specified position, deleting a node from the beginning of the list, and displaying the list. The code uses standard input/output operations like printf and scanf.

```
220     scanf("%d",&item);
221     start=insertlend(start,item);
222     display(start);
223     break;
224
225     case 3:
226         printf("Enter element to insert: ");
227         scanf("%d",&item);
228         printf("Enter position: ");
229         scanf("%d",&pos);
230         start = insertlpos(start, item, pos);
231         display(start);
232         break;
233     case 4:
234         start=deletesllbeg(start);
235         display(start);
236         break;
237
238     case 5:
239         start=deletesllend(start);
240         display(start);
241         break;
242
243     case 6:
244         printf("Enter position to delete: ");
245         scanf("%d",&pos);
246         start = deleteslpos(start, pos);
247         display(start);
248         break;
249
250     case 7: return 0;
251
252     default printf("Invalid input");
253
254 }
255
256 }
257 }
```

C:\Users\gupta\Desktop\yitu dsal\p3.c      C/C++      Windows (CR+LF)      WINDOWS-1252      Line 101, Col 20, Pos 2070      Insert      Modified      Read/Write      default      ENG IN      09:38      22-11-2025

The screenshot shows the same Code::Blocks 20.03 interface, but the code has been modified to include a default case and a return statement. The default case handles invalid input by printing "Invalid input". The program now returns 0 at the end of execution.

```
226     printf("Enter element to insert: ");
227     scanf("%d",&item);
228     printf("Enter position: ");
229     scanf("%d",&pos);
230     start = insertlpos(start, item, pos);
231     display(start);
232     break;
233
234     case 4:
235         start=deletesllbeg(start);
236         display(start);
237         break;
238
239     case 5:
240         start=deletesllend(start);
241         display(start);
242         break;
243
244     case 6:
245         printf("Enter position to delete: ");
246         scanf("%d",&pos);
247         start = deleteslpos(start, pos);
248         display(start);
249         break;
250
251     case 7: return 0;
252
253     default printf("Invalid input");
254
255 }
256
257 }
```

C:\Users\gupta\Desktop\yitu dsal\p3.c      C/C++      Windows (CR+LF)      WINDOWS-1252      Line 101, Col 20, Pos 2070      Insert      Modified      Read/Write      default      ENG IN      09:38      22-11-2025

## OUTPUT:

```
C:\Users\gupta\Desktop\ritu > + - 
Creation of linked list:
Enter element to be inserted (to stop enter -999): 2
3
4
5
-999
Linked List: 2 -> 3 -> 4 -> 5 -> NULL

*****MENU*****
1.Insert at the beginning
2.Insert at the end
3.Insert at a given position
4.Delete at the beginning
5.Delete at the end
6.Delete at the given position
7.Exit
1
Enter item to insert at beginning:1
Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL

*****MENU*****
1.Insert at the beginning
2.Insert at the end
3.Insert at a given position
4.Delete at the beginning
5.Delete at the end
6.Delete at the given position
7.Exit
2
Enter item to insert at end:6
Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> NULL

*****MENU*****

```

```
18°C Mostly clear Search ENG IN 09:44 22-11-2025
C:\Users\gupta\Desktop\ritu > + - 
*****MENU*****
1.Insert at the beginning
2.Insert at the end
3.Insert at a given position
4.Delete at the beginning
5.Delete at the end
6.Delete at the given position
7.Exit
3
Enter element to insert: 4
Enter position: 3
Linked List: 1 -> 2 -> 4 -> 3 -> 4 -> 5 -> 6 -> NULL

*****MENU*****
1.Insert at the beginning
2.Insert at the end
3.Insert at a given position
4.Delete at the beginning
5.Delete at the end
6.Delete at the given position
7.Exit
4
Deleted element from beginning is:1
Linked List: 2 -> 4 -> 3 -> 4 -> 5 -> 6 -> NULL

*****MENU*****
1.Insert at the beginning
2.Insert at the end
3.Insert at a given position
4.Delete at the beginning
5.Delete at the end
6.Delete at the given position
7.Exit
5

```

```
*C:\Users\gupta\Desktop\ritu * + - X
.2.Insert at the end
.3.Insert at a given position
.4.Delete at the beginning
.5.Delete at the end
.6.Delete at the given position
.7.Exit
5
Deleted element from end is:6
Linked List: 2 -> 4 -> 3 -> 4 -> 5 -> NULL

*****MENU*****
1.Insert at the beginning
.2.Insert at the end
.3.Insert at a given position
.4.Delete at the beginning
.5.Delete at the end
.6.Delete at the given position
.7.Exit
6
Enter position to delete: 2
Deleted element is 4
Linked List: 2 -> 3 -> 4 -> 5 -> NULL

*****MENU*****
1.Insert at the beginning
.2.Insert at the end
.3.Insert at a given position
.4.Delete at the beginning
.5.Delete at the end
.6.Delete at the given position
.7.Exit
7

Process returned 0 (0x0) execution time : 49.951 s
Press any key to continue.
|
```

