1	A
9-	# include (stdio.h)
J	Struct node of
)——	int info; Monet node * link;
)	Struct node * link;
	6
	typedol struct node *NODE;
	Note at nedo():
	8
	NODE X;
	X= (NOIDE) mallor (Size of (NOOE));
	$\frac{1}{\sqrt{x}} = NULL$
	de la companya de la
,	brint ( 'Momora Pall 50'):
	print ('Memory full on'); exit(o);
	b b
	neturn x;
	1
	Votid friende (NODE x)
	NODE insert front (NODE first, int litery);
	NODE insort front (NODE Ring - int Ditom) -
	& Daniel Marie
	NODE temp = getnode();
	temp - into = item =
	temb-> link = NULL;
	el (first = NULL)
	temp -> into = item =  temp -> link = NULL;  ef (first == NULL)  constitute temp;
	temp > tenk = first;

noturn temp;
+1
NODE delète-front (NODE fint)
1 1000 tamb
NODE temp;  El (first == NULL)
print ("List is empty \n");
neturn first;
print (" Item deleted «/od", (first >info));
temb = text;
tamb - 10mb -> VIIIK
free (first);
return temp;
NODE delete Tear (NODE first)
8
NODE TURE = first preus= NULL;
if (first == NULL)
d'
print ("List empty (n");
neturn NULL;
3
"  "  "  "  "  "  "  "  "  "  "  "  "
print the deleted is you it form sugo)
return mit; NULL;
To The same of the
While (cur > link = NULL) &
prove cure;
an = an slink; b

SPLASH
printf (" Storn doleted a god (1, cur - info);
paint (" Sitom deleted is % a 11, 11
fall (lun);
Drou->link = NDLL;
natura first;
2
NODE del pos (NODE first, int pos)
in auto (mar ban)
NICOC AND BOOK O
None cur, frapreu;
M (=1)
4 (first = = NOLL 11 pors < 0)
1 et un or el vite voul
print (" and Involed position \n");
return NULL;
<b>b</b>
4(pes == 1)
d'
free (first);
return NOLL;
3
Cur = first?
borou = RIVII ;
prov = KULI; while ( curi = NULL)
§
\$\(C==\pos\)
§ · · · · · · · · · · · · · · · · · · ·
point ("Floment deleted & Tod" cur >info);
brow -> link = an -> link;
free Europ;
b b
pray = arr;
au = au p -> link;

point ("F liment not found in"); vad daplay (NODE first mint ("l'ext a empty 10"); poliunis wint ("Flomonts of the list are; (a"); on (NOOE 9= first; 9 != NULL; 9=9 > link). coluit ("old in "9 >into); ut main ( int itom, ch, bes NOOE LINT = MULL; Printf ("\n 1. Prixtit \n2. delato front \n3. delato rom \n2. sunt ( " % d' & ch); Match (th) rax: paint ("Enter element to be inested in"); renef ("Yod", & "lim);

- first = "next front (first, "tem);

brook; cax 2: first = dolate-front (first);

broak

```
1 #include <stdio.h>
 2 #include <stdlib.h>
3 struct node{
4
   int info;
 5
   struct node *link;
 6 };
 7 typedef struct node *NODE;
8 NODE getnode()
9 {
10
   NODE x;
11
   x=(NODE)malloc(sizeof(NODE));//
12
   if(x==NULL)
13
   {
14
     printf("memory full \n");
15
    exit(0);
16
17
   return x;
18 }
19
20 void freenode(NODE x)
21 {
22
    free(x);
23 }
24
25 NODE insert_front(NODE first,int item)
26 {
27
   NODE temp = getnode();
28
   temp->info = item;
29
   temp->link = NULL;
30 if(first == NULL)
31
   return temp;
32
   temp->link=first;
33 return temp;
34 }
35
36 NODE delete_front(NODE first)
37 {
38
   NODE temp;
39
   if(first == NULL)
40
    printf("List is empty\n");
41
42
     return first;
43
   printf("Item deleted %d",(first->info));
44
45
   temp = first;
   temp=temp->link;
46
   free(first);
48
    return temp;
49 }
50
51 NODE delete_rear(NODE first)
52 {
53 NODE cur=first,prev=NULL;
```

```
54
     if(first==NULL)
 55
    {
      printf("List empty\n");
 56
 57
     return NULL;
 58
     }
 59
     if(first->link==NULL)
 60
       printf("item deleted is
 61
   %d\n",first->info);
      free(first);
 62
 63
      return NULL;
 64
 65
    while(cur->link!=NULL)
 66
 67
      prev=cur;
 68
     cur=cur->link;
 69
 70
     printf("Item deleted is %d\n",(cur->info));
 71
    free(cur);
 72
     prev->link=NULL;
 73
    return first;
 74 }
 75
 76 void display(NODE first)
 77 {
 78
    if(first==NULL)
 79
      printf("List is empty\n");
 80
 81
      return;
 82
 83
     printf("Elements of the list are : \n");
     for(NODE i=first;i!=NULL;i=i->link)
 84
 85
     printf("%d\n",i->info);
 86 }
 87
 88 NODE delete_pos(NODE first,int pos)
 89 {
 90
    NODE cur, prev;
 91
    int c=1;
 92
     if(first==NULL||pos<0)</pre>
 93
 94
      printf("Invalid position\n");
 95
      return NULL;
 96
 97
     if(pos==1)
 98
 99
      free(first);
100
      return NULL;
101
102
    cur=first;
103
     prev=NULL;
104
     while(cur!=NULL)
105
```

```
{
105
106
      if(c==pos)
       {
107
        printf("Element deleted is
108
   %d",cur->info);
        prev->link=cur->link;
109
        free(cur);
110
111
        return first;
112
      }
113
      prev=cur;
114
      cur=cur->link;
115
       C++;
     }
116
     printf("Element not found\n");
117
118
     return first;
119 }
120
121
122
123
124 int main()
125 {
126 int item, ch, pos;
     NODE first=NULL;
127
128
    for(;;)
     {
129
130
       printf("\n1.Insert front\n2.Delete
    front\n3.Delete
    rear\n4.Delete_pos\n5.Display\n");
       scanf("%d", &ch);
131
132
      switch(ch)
133
134
        case 1:
135
         printf("Enter element to be
    inserted\n");
         scanf("%d",&item);
136
137
         first = insert_front(first,item);
138
         break;
        case 2:
139
         first = delete_front(first);
140
141
         break:
142
        case 3:
143
         first = delete_rear(first);
144
          break;
145
        case 4:
```

```
146
          printf("Enter position\n");
147
          scanf("%d",&pos);
          first = delete_pos(first,pos);
148
149
          break;
        case 5:
150
          display(first);
151
152
          break;
153
        default:return 0;
154
155
156 }
```

```
1.Insert front
Delete front
Delete rear
4.Delete_pos
5.Display
Enter element to be inserted
10
1. Insert front
2.Delete front
3.Delete rear
4.Delete_pos
5.Display
Enter element to be inserted

    Insert front

Delete front
3.Delete rear
4.Delete_pos
5.Display
Enter element to be inserted
30
1.Insert front
Delete front
3.Delete rear
4.Delete_pos
5.Display
Elements of the list are :
40
30
20
10
1. Insert front
Delete front
3.Delete rear
4.Delete pos
5.Display
Item deleted 40

    Insert front

2.Delete front
```

```
Item deleted 40

    Insert front

2.Delete front
Delete rear
4.Delete_pos
5.Display
Elements of the list are :
30
20
10

    Insert front

Delete front
Delete rear
4.Delete_pos
5.Display
Item deleted is 10

    Insert front

Delete front
Delete rear
4.Delete_pos
Display
Elements of the list are :
30
20

    Insert front

Delete front
3.Delete rear
4.Delete_pos
5.Display
Enter position
Element deleted is 20
1.Insert front
2.Delete front
Delete rear
4.Delete_pos
5.Display
Elements of the list are :
30
```