```
#include<stdio.h>
#include<stdlib.h>
 struct node
{
    int data;
   struct node *next;   struct node * prev
};
 void struct node * head;
 void beginsert();
 void bndinsrt();
 void randinsert();
 void begdel();
 void enddel();
 void randdel();
 void main()
{
  int choice=0;
  while(choice!=8)
{
  printf("\n1. insert at the beginning \n2. Insert at the en
3. Insert at random \n4. delete at the beginning \n5. dele
at the end \n6. Delete at random \n7. Display \n8. Exit
  printf("Enter any choice: ");
  scanf("%d",&choice);
  switch(choice)
{
  case 1: begininsert();
  break;
  case 2: bndinsert();
   break;
  case 3: randinsert();
```

```
break;
case 4: begdel();
 break;
case 5: enddel();
 break;
case 6: randdel();
 break;
case 7: display();
 break;
case 8; Exit(0);
 break;
}
}


void beginsert()
{
struct node *ptr;
int item;
ptr = (struct node *) malloc(sizeof(struct node));
if(ptr == NULL)
{
  printf("\n overflow");
} else {
  printf("\n Enter item value");
  scanf("%d", &item);
  if(head == NULL)
  {
  ptr -> next = NULL;
  ptr -> prev = NULL;
  ptr -> data = item;
  head = ptr;
}
```

```
else
{ ptr -> data = item;
  ptr -> prev = NULL;
  ptr -> next = head;
  head -> prev = ptr;
  head = ptr;
} printf("\n node inserted");
 }
{
 void endinsert()
{ struct node *ptr, *temp;
 int item;
 ptr = (struct node *)malloc(size of (struct node));
  if(ptr == NULL)
{ printf("\n overflow);
} else
{ printf("\n enter value");
  scanf("%d", &item);
   ptr -> data = item;
 if(head == NULL)
{  ptr -> next = NULL;
    ptr -> prev = NULL;
    head = ptr;
} else {
 temp = head;
 while(temp -> next != NULL)
 { temp = temp -> next;
 }
 temp -> next = ptr;
 ptr -> next = NULL;
 printf("\n node inserted");
3 }
```

store
67

```
void randinsert()
{
  int item,loc,i;
  struct node *ptr,*temp;
  ptr=(struct node*)malloc(size of(struct node));
  if(ptr==NULL)
  { printf("Overflow");
  }else
  { temp=head;
    printf("Enter location:");
    scanf("%d",&loc);
    for(i=0;i<loc;i++)
    { temp=temp->next;
      if(temp==NULL)
      { printf("there are less than %d elements",loc)
        return;
      }
    }
    printf("Enter value");
    scanf("%d",&item);
    ptr->data=item;
    ptr->next=temp->next;
    ptr->prev=temp;
    temp->next=ptr;
    temp->next->prev=ptr;
    printf("\n node inserted\n");
  }
}
void begdel()
{ struct node *ptr;
  if(head==NULL)
  { printf("underflow");
  }
```

```
else if (head->next == NULL)
{ head=NULL;
  free(head);
  printf("\n node deleted\n");
} else
{ ptr = head;
  head = head->next;
  head->prev = NULL;
  free(ptr);
  printf("\n node deleted");
}}
void delend()
{ struct node *ptr;
  if(head == NULL)
  { printf("\n underflow");
  } else if (head->next == NULL)
  { head = NULL;
    free(head);
    printf("\n node deleted\n");
  } else
  { ptr = head;
    if(ptr->next != NULL)
    { ptr = ptr->next;
    } ptr->prev->next = NULL;
  free(ptr);
  printf("\n node deleted\n");
}}
void randdel()
{ struct node *ptr, *temp;
  int val;
  printf("Enter the location: ");
  scanf("%d", &val);
  ptr = head;
```

```
while(ptr -> data != val)
  ptr = ptr -> next;
  if (ptr -> next == NULL)
  {printf("Can't delete");
  }

  else if(ptr -> next -> next == NULL)
  {ptr -> next = NULL;
  }else
  {temp = ptr -> next;
   ptr -> next = temp -> next;
   temp -> next -> prev = ptr;
   free(temp);
   printf("\n node deleted\n");
}}

void display()
{ struct node *ptr;
  printf("printing value :\n");
    ptr = head;
    while(ptr != NULL)
  { printf("%d\n", ptr -> data)
    ptr = ptr -> next;
  }
}
```