**Single linked list**

#include<iostream>

#include<malloc.h>

using namespace std;

struct node

{

   int data;

   struct node \*next;

}\*list=NULL,\*p,\*q,\*r;

class linked

{

   public:

      int data1,data2;

   void menu()

   {

      int ch;

      do

      {

          cout<<endl<<"enter your choice \n 1.Insert at beginning \n2.Display \n 3. insert at the end \n4. insert before a particular node\n5. insert after a particular node\n 7. Exit\n8. delete from beggining\n9. delete from end \n10 delete particular\n11 sort list\n12 Count the number of elements"<<endl;

          cout<<"13 Reverse of the linked list\n\t Enter your choice: ";

          cin>>ch;

          switch(ch)

          {

              case 1:

                  insertb();

                  break;

              case 2:

                  display();

                  break;

              case 3:

                  inserte();

                  break;

              case 4:

                  insertbap();

                  break;

              case 5:

                  insertaap();

                  break;

              case 7:

                  break;

              case 8:

                  deletefrombeg();

                  break;

              case 9:

                  deletefromend();

                  break;

              case 10:

                  deletepa();

                  break;

              case 11:

                  sortlist();

                  break;

              case 12:

                  cnt();

                  break;

              case 13:

                  rev();

                  break;

          }

      }while(ch!=7);

   }

   void cnt()

   {

      if(list==NULL)

      {

          cout<<"\n0 elements in the list";

      }

      else

      {

          q=list;

          int cnt=0;

          while(q!=NULL)

          {

              cnt=cnt+1;

              q=q->next;

          }

          cout<<"\n Number of elements in the linked list= "<<cnt;

      }

   }

   void rev()

   {

      if(list==NULL)

      {

          cout<<"\nList is empty";

      }

      else

      {

              p=NULL;

              q=list;

              while(q!=NULL)

              {

                  r=q->next;

                  q->next=p;

                  p=q;

                  q=r;

              }

              list=p;

      }

   }

   void sortlist()

   {

      if(list==NULL)

      {

          cout<<"\nList is empty";

      }

      else

      {

          q=list;

          while(q!=NULL)

          {

              r=q->next;

              while(r!=NULL)

              {

                  if(q->data > r->data)

                  {

                      data2=q->data;

                      q->data=r->data;

                      r->data=data2;

                  }

                  r=r->next;

              }

              q=q->next;

          }

      }

   }

   void deletefrombeg()

   {

      if(list==NULL)

      {

          cout<<"\nList is empty. can not delete element";

      }

      else

      {

          q=list;

          list=list->next;

          cout<<"\nElement "<<q->data<<" is deleted.";

          free(q);

      }

   }

   void deletefromend()

   {

      if(list==NULL)

      {

          cout<<"\nList is empty. can not delete element";

      }

      else

      {

          if(list->next==NULL)

          {

              cout<<"\nElement "<<list->data<<" List is deleted";

              list=NULL;

          }

          else

          {

          q=list;

          while(q->next!=NULL)

          {

              r=q;

              q=q->next;

          }

          cout<<"\nElement "<<q->data<<" is deleted";

          r->next=NULL;

          free(q);

          }

      }

   }

   void deletepa()

   {

      if(list==NULL)

      {

          cout<<"\nList is empty. can not delete element";

      }

      else

      {

          cout<<"\nWhich element you want to delete?";

          cin>>data2;

          q=list;

          while(q->data!=data2 && q->next!=NULL)

          {

              r=q;

              q=q->next;

          }

          if(q->data==data2)

          {

              r->next=q->next;

              free(q);

          }

          else

          {

              cout<<"\nElement is not found";

          }

      }

   }

void insertaap()

{

   if(list==NULL)

   {

      cout<<"\nCannot insert new value";

   }

   else

   {

      p=(struct node\*)malloc(sizeof(node));

      cout<<endl<<"enter data"<<endl;

        cin>>data1;

      p->data=data1;

      cout<<"\nEnter data after which you want insert new value";

      cin>>data2;

      q=list;

      while(q->data!=data2 && q->next!=NULL)

      {

          q=q->next;

      }

      if(q->data==data2)

      {

          r=q->next;

          q->next=p;

          p->next=r;

      }

      else

      {

          cout<<"\nData not found";

      }

   }

}

void insertbap()

{

   if(list==NULL)

   {

      cout<<"\nCannot insert new value";

   }

   else

   {

      p=(struct node\*)malloc(sizeof(node));

      cout<<endl<<"enter data"<<endl;

        cin>>data1;

      p->data=data1;

      cout<<"\nEnter data before which you want insert new value";

      cin>>data2;

      q=list;

      while(q->data!=data2 && q->next!=NULL)

      {

          r=q;

          q=q->next;

      }

      if(q->data==data2)

      {

          r->next=p;

          p->next=q;

      }

      else

      {

          cout<<"\nData not found";

      }

   }

}

   void inserte()

   {

      if(list==NULL)

      {

          p=(struct node\*)malloc(sizeof(node));

          cout<<endl<<"enter data"<<endl;

            cin>>data1;

          p->data=data1;

          p->next=NULL;

          list=p;

      }

      else

      {

          p=(struct node\*)malloc(sizeof(node));

          cout<<endl<<"enter data"<<endl;

            cin>>data1;

          p->data=data1;

          q=list;

          while(q->next!=NULL)

          {

              q=q->next;

          }

          q->next=p;

          p->next=NULL;

      }

   }

   void insertb()

   {

      if(list==NULL)

      {

          p=(struct node\*)malloc(sizeof(node));

          cout<<endl<<"enter data"<<endl;

            cin>>data1;

          p->data=data1;

          p->next=NULL;

          list=p;

      }

      else

      {

              p=(struct node\*)malloc(sizeof(node));

          cout<<endl<<"enter data"<<endl;

            cin>>data1;

          p->data=data1;

          p->next=list;

          list=p;

      }

   }

   void display()

   {

      if(list==NULL)

      {

          cout<<endl<<"List is empty :( "<<endl;

      }

      else

      {

          q=list;

          while(q!=NULL)

          {

      cout<< q->data<<"---->";

              q=q->next;

          }

      }

   }

};

int main()

{

   linked l;

   l.menu();

   return 0;

}

**Output :**

**At beginning...**

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 1

enter data

10

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 1

enter data

20

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 1

enter data

30

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

30---->20---->10---->

**Insertion at the end of linked list**

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 3

enter data

88

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

30---->20---->10---->88---->

**Insert Before a particular Node.**

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 4

enter data

11

Enter data before which you want insert new value88

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

30---->20---->10---->11---->88---->

**Insertion After a Particular Element**

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 5

enter data

13

Enter data after which you want insert new value20

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

30---->20---->13---->10---->11---->88---->

**Delete Element from the beginning**

30---->20---->13---->10---->11---->88---->

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 8

Element 30 is deleted.

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

20---->13---->10---->11---->88---->

**Delete element at the End of linked list**

20---->13---->10---->11---->88---->

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 9

Element 88 is deleted

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

20---->13---->10---->11---->

**Delete a particular Element from the linked list**

20---->13---->10---->11---->

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 10

Which element you want to delete?10

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

20---->13---->11---->

**Sort a linked list**

20---->13---->11---->

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 11

enter your choice

1.Insert at beginning

2.Display

3. insert at the end

4. insert before a particular node

5. insert after a particular node

7. Exit

8. delete from beggining

9. delete from end

10 delete particular

11 sort list

12 Count the number of elements

13 Reverse of the linked list

    Enter your choice: 2

11---->13---->20---->