NUST COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

EC 220: COMPUTER SYSTEM ARCHITECTURE

LINKELIST CLASS

Instructor: Anum Abdul Salam

Lab Engineer: Hafsa

Student's Name: **Furqan Ahmad**

Reg Number: **352076**

Syndicate: **A**

Degree: **42**

Department: **Computer Engineering**

MAIN.CPP:

```
#include<iostream>
#include"Linkedlist.h"
using namespace std;

void objb(linkedlist &a){
       linkedlist b(a);
       while (true){
start:
              int x;

              cout<<"Choe the below options:\n";
              cout<<"1->Insert at start          2->Insert at End           3->Insert at
any Position\n";
              cout<<"4->Delete at start          5->Delete at End           6->Delete at
any Position\n";
              cout<<"7->Display                  8->Size                           0-
>Exit\n";
              cout<<"Choose:";
              cin>>x;
              switch(x){
                     case 1:
                            cout<<"Insert at start:\n";
                            b.instrt();
                            system("pause");
                            break;
                     case 2:
                            cout<<"Insert at End:";
                            b.insend();
                            system("pause");
                            break;
                     case 3:
                            cout<<"Insert at any Position:\n";
                            b.insatanyp();
                            system("pause");
                            break;
                     case 4:
                            cout<<"Delete at start:";
                            b.delstrt();
                            system("pause");
                            break;
                     case 5:
                            cout<<"Delete at End:";
                            b.delend();
                            system("pause");
                            break;
                     case 6:
                            cout<<"Delete at any Position:\n";
                            b.delatanyp();
                                   system("pause");
                            break;
                     case 7:
                            cout<<"Display:\n";
                            b.display();
                            system("pause");
                            break;
```

```
                        case 8:
                                cout<<"Size: "<<b.size()<<endl;
                                system("pause");
                                break;
                        case 0:
                                a.~linkedlist();
                                exit(0);
                                break;
                        default:
                                system("cls");
                                goto start;


                }
                system("cls");
        }

}


int main(){
        linkedlist a;
        a.instrt();
        a.instrt();
        a.instrt();
        a.instrt();
        cout<<a.size()<<endl;
        while (true){
start:
                int x;

                cout<<"Choe the below options:\n";
                cout<<"1->Insert at start          2->Insert at End            3->Insert at
any Position\n";
                cout<<"4->Delete at start          5->Delete at End            6->Delete at
any Position\n";
                cout<<"7->Display                  8->Size                            0-
>Exit\n";
                cout<<"9->Copy Object\n";
                cout<<"Choose:";
                cin>>x;
                switch(x){
                        case 1:
                                cout<<"Insert at start:\n";
                                a.instrt();
                                system("pause");
                                break;
                        case 2:
                                cout<<"Insert at End:";
                                a.insend();
                                system("pause");
                                break;
                        case 3:
                                cout<<"Insert at any Position:\n";
                                a.insatanyp();
                                system("pause");
                                break;
                        case 4:
```

```
                cout<<"Delete at start:";
                a.delstrt();
                system("pause");
                break;
            case 5:
                cout<<"Delete at End:";
                a.delend();
                system("pause");
                break;
            case 6:
                cout<<"Delete at any Position:\n";
                a.delatanyp();
                    system("pause");
                break;
            case 7:
                cout<<"Display:\n";
                a.display();
                system("pause");
                break;
            case 8:
                cout<<"Size: "<<a.size()<<endl;
                system("pause");
                break;
            case 9:
                objb(a);
                break;
            case 0:
                a.~linkedlist();
                exit(0);
                break;
            default:
                system("cls");
                goto start;


        }
        system("cls");
    }
    system("pause");
return 0;
}
```

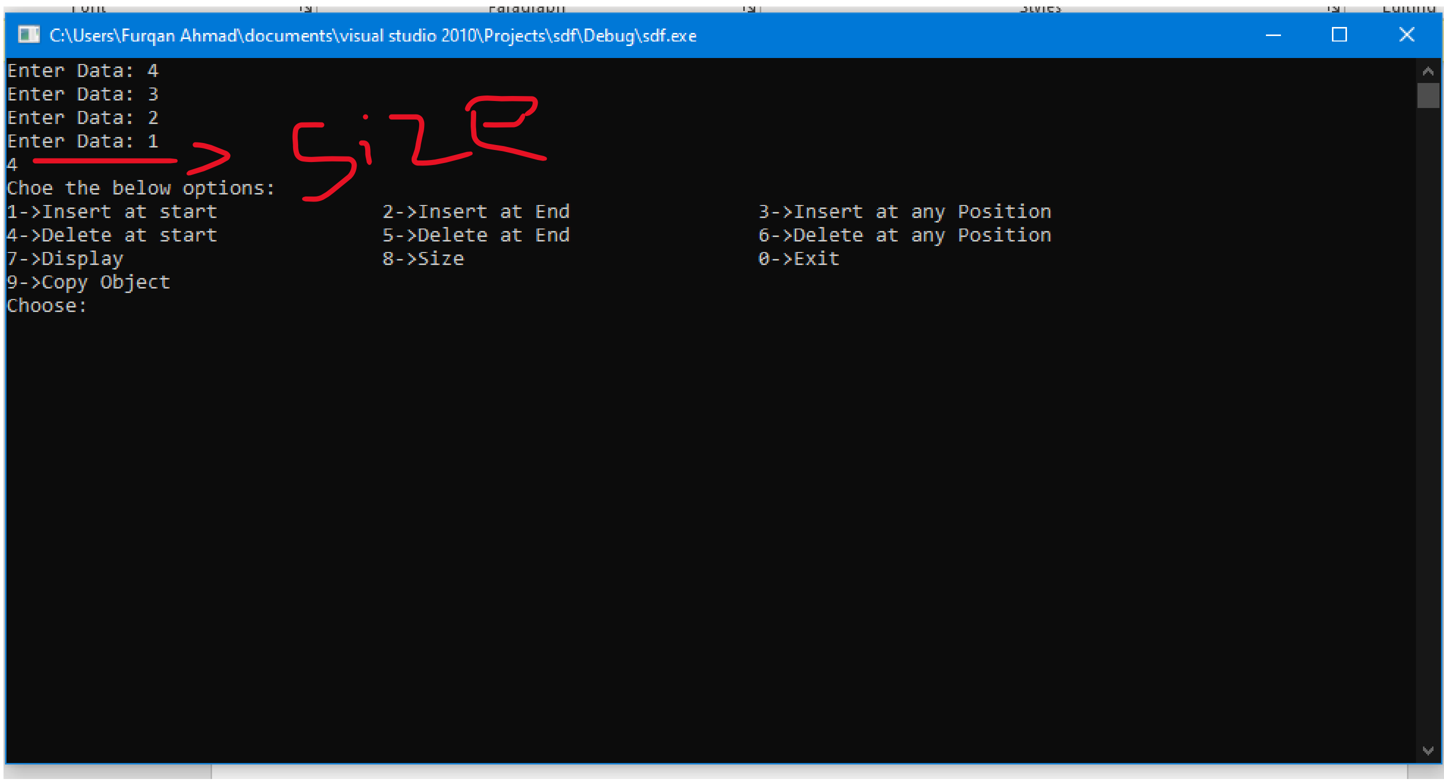
HEADER.H:

```
#include<iostream>
using namespace std;
struct node{
int val;
node *add;
};
///////////////////////////////////////////////////////////////////

class linkedlist{
node *head;
public:
        linkedlist();
        linkedlist(linkedlist &b);
        void instrt();
        void insend();
        void insatanyp();
        void delstrt();
        void delend();
        void delatanyp();
        void display();
        int size();
        bool isempty();
        ~linkedlist();
        };


///////////////////////////////////////////////////////////////////


//Constructor
        linkedlist::linkedlist(){
        head=NULL;
        }

///////////////////////////////////////////////////////////////////

        //insert at start
        void linkedlist::instrt(){
                node *a=new node;

        if(isempty()){
        cout<<"Enter Data: ";
        cin>>a->val;
        head=a;
        a->add=NULL;
        }
        else{
        cout<<"Enter Data: ";
        cin>>a->val;
        a->add=head;
        head=a;
        }
        }
```

```
///////////////////////////////////////////////////////////////////

       //insert at end
       void linkedlist::insend(){
              node *a=new node;
       node *temp;
       temp = head;

       while(true){
              if(temp->add==NULL){
       cout<<"Enter Data: ";
       cin>>a->val;
              temp->add=a;
              a->add=NULL;
              break;
              }
              else{temp=temp->add;}
       }

       }

///////////////////////////////////////////////////////////////////

       //delete at start
       void linkedlist::delstrt(){
       node *temp=head;
       head=head->add;
       delete temp;
       }
///////////////////////////////////////////////////////////////////


       //delete at end
       void linkedlist::delend(){
       node *temp;
       temp = head;


       while(true){
              node * temp2=temp->add;
              if(temp2->add==NULL){
              delete temp2;
              temp->add=NULL;
              break;
              }
              else{temp=temp->add;

              }
       }
       }
///////////////////////////////////////////////////////////////////

       //display
       void linkedlist::display(){
       node *temp;
       temp = head;

       while(temp!=NULL){
```

```
                cout<<"Value "<<temp->val<<endl;
                temp=temp->add;
        }

        }

////////////////////////////////////////////////////////////////////

        //isempty
        bool linkedlist::isempty(){
        if(head==NULL)
                return true;
        else
                return false;
        }
////////////////////////////////////////////////////////////////////

        //Size
        int linkedlist::size(){
        int count=1;
        if(isempty()){return 0;}
        node *temp=head;
        while(temp->add!=NULL){
                temp=temp->add;
        count++;}
        return count;
        }
////////////////////////////////////////////////////////////////////

        //inert at any position
        void linkedlist::insatanyp(){
                int x;
                cout<<"Enter your position: ";
                cin>>x;

                if(x>size()||x<1){cout<<"This Position Does not Exist!\n";}

                if(x==1){instrt();}


                if(x>1&&x<size()){
                node *temp,*temp2;
                temp=head;
                node *a=new node;
                temp2=temp->add;
                int count=1;

                while(true){
                        count++;
                if(count==x){cout<<"Enter value: ";
                cin>>a->val;
                temp->add=a;
                a->add=temp2;
                break;}
                temp=temp->add;
                        temp2=temp2->add;
                }
                }
                if(x==size()){

                node *temp;
        temp = head;
        node *a=new node;

        while(true){
                node * temp2=temp->add;
                if(temp2->add==NULL){
                cout<<"Enter Value: ";
                cin>>a->val;
                a->add=temp2;
                temp->add=a;
                break;
                }
                else{temp=temp->add;

                }
        }}


        }



///////////////////////////////////////////////////////////////////

        //delete at any position
void linkedlist::delatanyp(){
int x;
                cout<<"Enter your position: ";
                cin>>x;

                if(x>size()||x<1){cout<<"This Position Does not Exist!\n";}

                if(x==1){delstrt();}


                if(x>1&&x<size()){
                node *temp,*temp2;
                temp=head;

                temp2=temp->add;
                int count=1;

                while(true){
                        count++;
                if(count==x){
                temp->add=temp2->add;
                delete temp2;
                temp2=NULL;
                break;}
                temp=temp->add;
                        temp2=temp2->add;
                }
                }
                if(x==size()){

                node *temp;
        temp = head;;

        while(true){
                node * temp2=temp->add;
                if(temp2->add==NULL){


                temp->add=temp2->add;
                delete temp2;
                temp2=NULL;
                break;
                }
                else{temp=temp->add;

                }
        }}


}
///////////////////////////////////////////////
//copy constructor

linkedlist::linkedlist(linkedlist &a){
        if(a.head==NULL){return;}
        head=nullptr;
node *temp=a.head;
node *temp2=nullptr;
node *temp3;
while(temp!=NULL){
        temp2=new node;
        int x=temp->val;
        temp2->val=temp->val;
        temp2->add=temp->add;
        if(this->head==NULL){
        head=temp2;
        temp3=head;}
        temp=temp->add;
}
}

////////////////////////////////////////////////////////////////

        //distructor
        linkedlist::~linkedlist(){
                while(true){
                if(!isempty()){
                        delend();}
                else
                        break;
                }
        }
//////////////////////////////////////////////////////
```

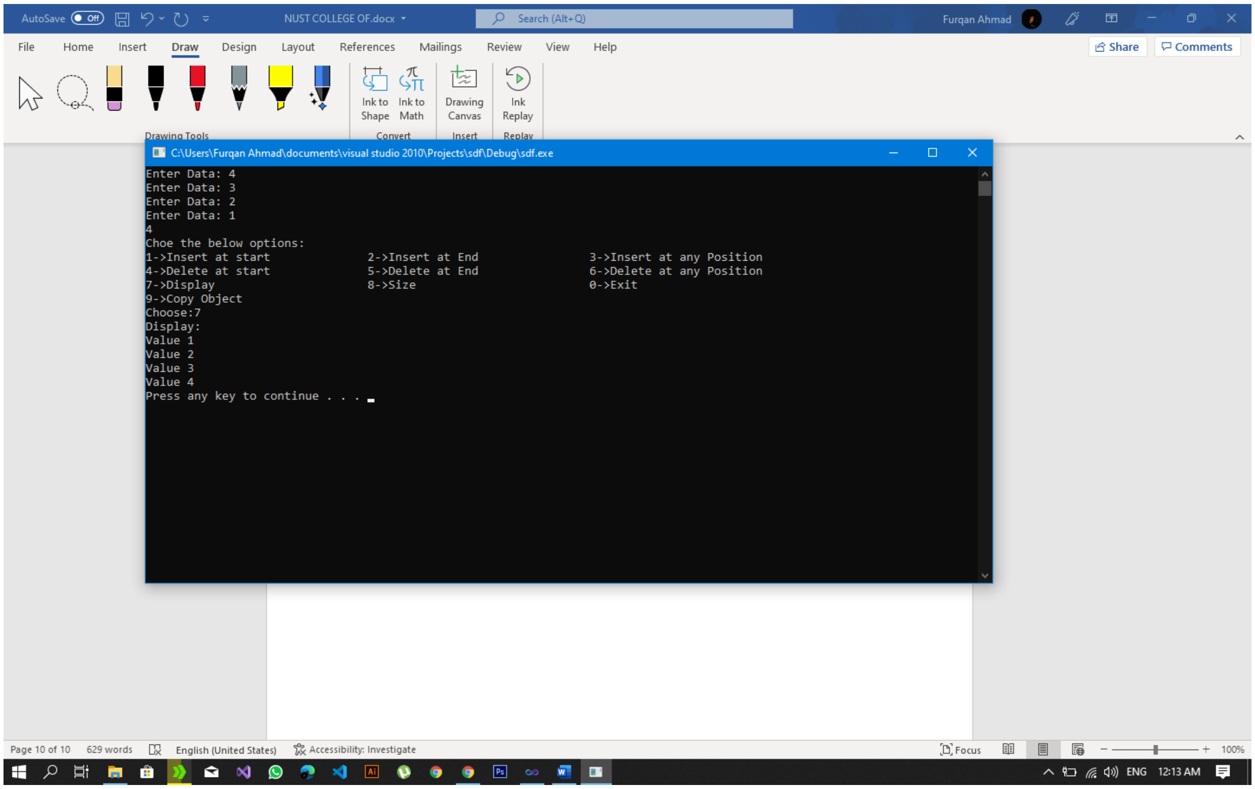
SCREENSHOT:



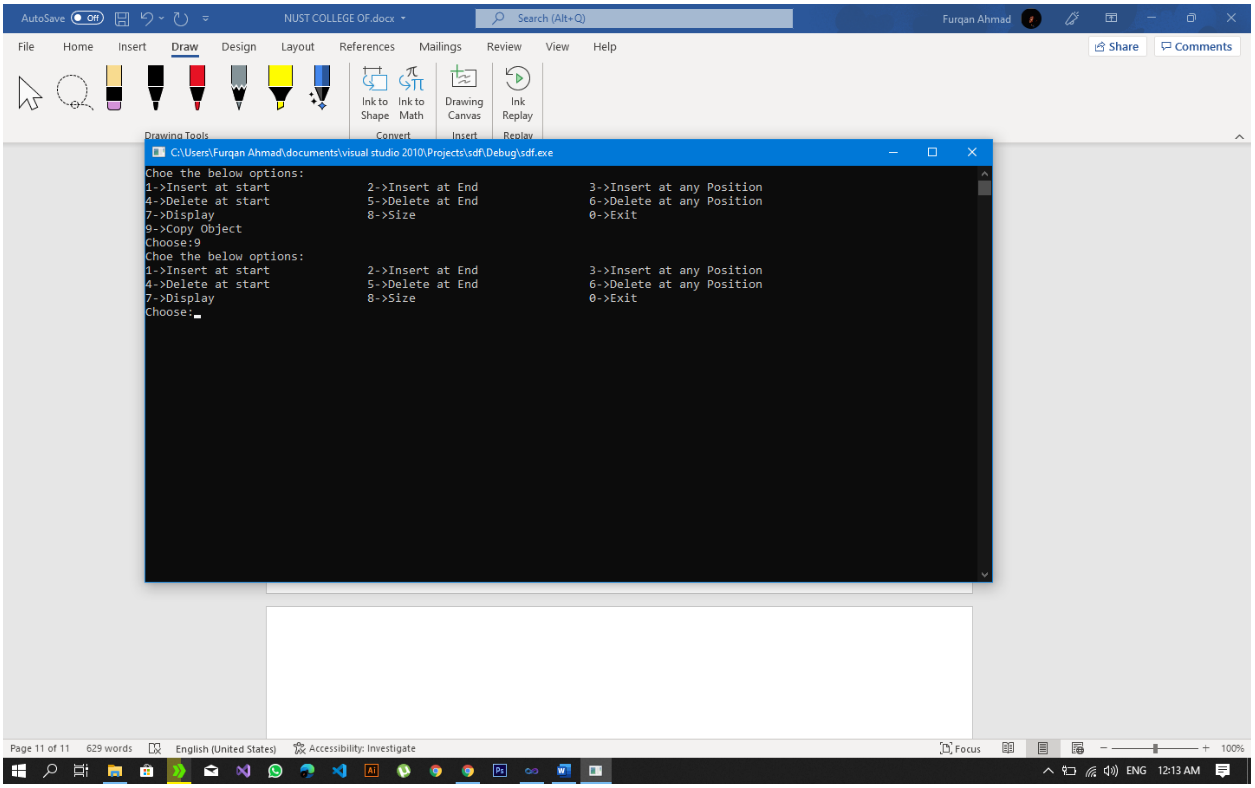
C:\Users\Furqan Ahmad\documents\visual studio 2010\Projects\sdf\Debug\sdf.exe

```
Enter Data: 4
Enter Data: 3
Enter Data: 2
Enter Data: 1
4
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
9->Copy Object
Choose:
```

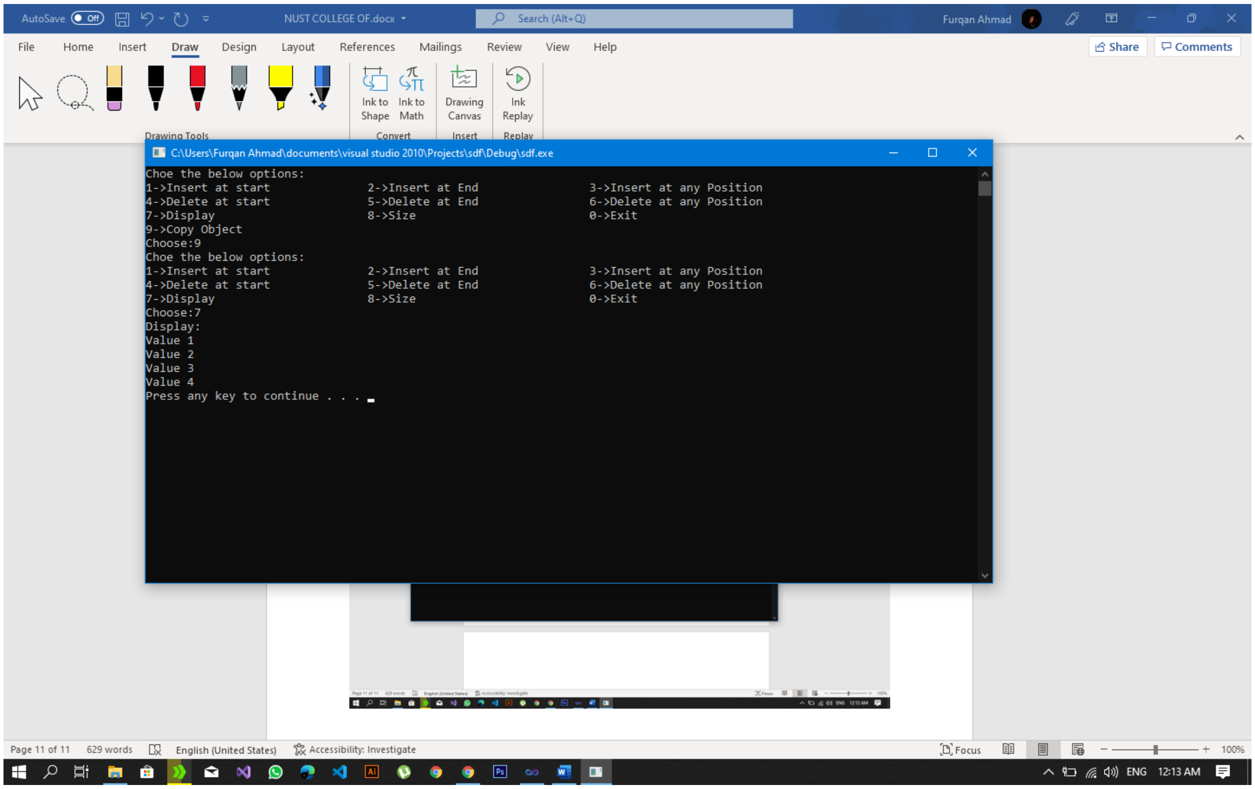
Size



```
Enter Data: 4
Enter Data: 3
Enter Data: 2
Enter Data: 1
4
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
9->Copy Object
Choose:7
Display:
Value 1
Value 2
Value 3
Value 4
Press any key to continue . . .
```


AutoSave Off
NUST COLLEGE OF.docx
Search (Alt+Q)
Furqan Ahmad
File Home Insert Draw Design Layout References Mailings Review View Help
Share Comments
Ink to Shape Ink to Math Drawing Canvas Ink Replay
Drawing Tools Convert Insert Replay
C:\Users\Furqan Ahmad\documents\visual studio 2010\Projects\sdf\Debug\sdf.exe
Page 11 of 11 629 words English (United States) Accessibility: Investigate
Focus 100%
ENG 12:13 AM

```
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
9->Copy Object
Choose:9
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
Choose:_
```


AutoSave Off
NUST COLLEGE OF.docx
Search (Alt+Q)
Furqan Ahmad
File Home Insert Draw Design Layout References Mailings Review View Help
Share Comments
Ink to Shape Ink to Math Drawing Canvas Ink Replay
Drawing Tools Convert Insert Replay
C:\Users\Furqan Ahmad\documents\visual studio 2010\Projects\sdf\Debug\sdf.exe
Page 11 of 11 629 words English (United States) Accessibility: Investigate
Focus 100%
ENG 12:13 AM

```
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
9->Copy Object
Choose:9
Choe the below options:
1->Insert at start              2->Insert at End                3->Insert at any Position
4->Delete at start              5->Delete at End                6->Delete at any Position
7->Display                      8->Size                         0->Exit
Choose:7
Display:
Value 1
Value 2
Value 3
Value 4
Press any key to continue . . . _
```

EVERY TASK IS WORKING!

Due to lake of time few screenshots are attached!