

```

C:\Users\Mark\Desktop\link list\LinkList
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 1
How many node do you want?: 3
Enter the elements: 2
Enter the elements: 3
Enter the elements: 4
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
2
3
4
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 2
Enter a value: 1
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
1
2
3
4
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: _

```

```

#include <iostream>
using namespace std;

//initializing the functions
void createlist(int data);
void display();
void addB(int data);
void addE(int data);
void delB();
void delE();

//worktool
struct Node {
    int info;
    struct Node *link;
    Node* prev;
}*start;

int main (){
    int choice,n,elements,i;
    // MENU
    while (1){
        cout << " 1 - Create a List \n";
        cout << " 2 - Add at the beginning\n";
        cout << " 3 - Add at the end\n";
        cout << " 4 - Delete the beginning\n";
        cout << " 5 - Delete the end\n";
        cout << " 6 - Display\n";
        cout << " 7 - EXIT\n";

        cout << "Enter your choice: ";
        cin >> choice;

        switch(choice)
        {
            case 1:
                cout << "How many node do you
want?: ";

                cin >> n;
                for(i=0;i<n;i++){
                    cout << "Enter the
elements: ";

                    cin >> elements;
                    createlist(elements);
                }
            }
        }
    }
}

```

```

1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 3
Enter a value: 5
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
1
2
3
4
5

1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 4
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
2
3
4
5

1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
2
3
4
5

```

```

    }
    break;
    case 2:
        cout << "Enter a value: ";
        cin >> elements;
        addB(elements);
        break;
    case 3:
        cout << "Enter a value: ";
        cin >> elements;
        addE(elements);
        break;
    case 4:
        delB();
        break;
    case 5:
        delE();
        break;
    case 6:
        display();
        break;
    case 7:
        return 0;
    }
}

void createlist(int data){
    // this will create a list
    struct Node *q, *tmp;
    tmp = new Node;
    tmp->info = data;
    tmp->link = NULL;
    if(start == NULL)
        start = tmp;
    else{
        q = start;
        while(q->link!=NULL)
            q=q->link;
        q->link = tmp;
    }
}

void display(){
    //this will display the datas
    struct Node *q;
    if (start == NULL)
        cout << "It's Empty!";
}

```

```

1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 5
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 6
2
3
4
1 - Create a List
2 - Add at the beginning
3 - Add at the end
4 - Delete the beginning
5 - Delete the end
6 - Display
7 - EXIT
Enter your choice: 7
-----
Process exited after 278.4 seconds with
Press any key to continue . . .

```

```

else
q = start;
while (q!=NULL){
    cout << q->info << endl;
    q=q->link;
}
cout << endl;
}

void addB(int data){
    // this will add a value on the beginning
    struct Node *head;
    head = new Node;
    head->info=data;
    if(start == NULL) {
        head->link = NULL;
    }
    else
        head->link = start;
    start = head;
}

void addE(int data){
    //this will add a value at the end
    struct Node *tail, *temp;
    tail = new Node;
    temp = new Node;
    tail->info=data;
    tail->link=NULL;
    if(start == NULL)
        start = new Node;
    else
        temp = start;
    while (temp->link!=NULL)
        temp = temp->link;
    temp->link = tail;
}

void delB(){
    struct Node *temp;
    temp = new Node;
    // moving the first value of the start to the 2nd
    // then making the temp the first value
    temp = start ;
    start = start->link;
    delete temp;
}

```

```
}  
void delE(){  
    struct Node *temp;  
    temp = new Node;  
    //initializing the end and start  
    Node *a = start;  
    Node *b = NULL;  
    while (a->link){  
        b = a;  
        a = a->link;  
    }  
    //making the a NULL  
    if (b){  
        b -> link = NULL;  
    }  
    delete a;  
}
```





