***#include <iostream>***

***#include <stdio.h>***

***#include <conio.h>***

***#include <stdlib.h>***

***Using namespace std;***

***Struct node***

***{***

***Int data;***

***Struct node \*next;***

***};***

***Struct node \*head;***

***Void insertion\_begin();***

***Void insertion\_sl();***

***Void insertion\_end();***

***Void deletion\_begin();***

***Void deletion\_sl();***

***Void deletion\_end();***

***Void display();***

***Void search();***

***Int main()***

***{***

***Int choice;***

***Char c;***

***Do***

***{***

***Cout << “\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n”;***

***Cout << “\n\t\t\t\t------Menu-----\n”;***

***Cout << “\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n”;***

***Cout << “\n\t1.Insertion at beginning”;***

***Cout << “\n\t2.Insertion at specific location”;***

***Cout << “\n\t3.Insertion at ending”;***

***Cout << “\n\t4.Deletion at beginning”;***

***Cout << “\n\t5.Deletion at specific location”;***

***Cout << “\n\t6.Deletion at ending”;***

***Cout << “\n\t7.Display”;***

***Cout << “\n\t8.Searching”;***

***Cout << “\nEnter your choice: “;***

***Cin >> choice;***

***Switch (choice)***

***{***

***Case 1:***

***Insertion\_begin();***

***Break;***

***Case 2:***

***Insertion\_sl();***

***Break;***

***Case 3:***

***Insertion\_end();***

***Break;***

***Case 4:***

***Deletion\_begin();***

***Break;***

***Case 5:***

***Deletion\_sl();***

***Break;***

***Case 6:***

***Deletion\_end();***

***Break;***

***Case 7:***

***Display();***

***Break;***

***Case 8:***

***Search();***

***Break;***

***Default:***

***Cout << “\nEnter a valid choice!”;***

***}***

***Cout << “\nDo you want to continue?(Y/N): “;***

***Cin >> c;***

***} while (c == ‘Y’ || c == ‘y’);***

***Return 0;***

***}***

***Void insertion\_begin()***

***{***

***Int item;***

***Struct node \*ptr = (struct node\*)malloc(sizeof(struct node));***

***Cout << “\nEnter an element:”;***

***Cin >> item;***

***Ptr->data = item;***

***If (head==NULL)***

***{***

***Head=ptr;***

***Head->next = NULL;***

***}***

***Else***

***{***

***Ptr->next = head;***

***Head = ptr;***

***}***

***}***

***Void insertion\_sl()***

***{***

***Int item,loc,I;***

***Struct node \*ptr,\*temp;***

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

***Cout << “\nEnter an element:”;***

***Cin >> item;***

***Ptr->data = item;***

***Temp=head;***

***If (head == NULL)***

***{***

***Head=ptr;***

***Head->next = NULL;***

***}***

***Else***

***{***

***Cout << “\nEnter after which loc to be inserted:”;***

***Cin >> loc;***

***Ptr->data=item;***

***For(i=1;i<=loc;i++)***

***{***

***Temp=temp->next;***

***If(temp==NULL)***

***{***

***Return;***

***}***

***}***

***Ptr->next=temp->next;***

***Temp->next=ptr;***

***}***

***}***

***Void insertion\_end()***

***{***

***Int item;***

***Struct node \*ptr,\*temp;***

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

***Cout << “\nEnter an element:”;***

***Cin >> item;***

***Ptr->data = item;***

***Temp=head;***

***If (head == NULL)***

***{***

***Head=ptr;***

***Head->next = NULL;***

***}***

***Else***

***{***

***While(temp->next!=NULL)***

***{***

***Temp=temp->next;***

***}***

***Temp->next=ptr;***

***Ptr->next=NULL;***

***}***

***}***

***Void deletion\_begin()***

***{***

***Struct node \*ptr;***

***If(head->next==NULL)***

***{***

***Ptr=head;***

***Head=NULL;***

***Free(ptr);***

***}***

***Else***

***{***

***Ptr=head;***

***Head=ptr->next;***

***Free(ptr);***

***}***

***}***

***Void deletion\_sl()***

***{***

***Struct node \*ptr,\*ptr1;***

***Int loc;***

***Cout << “\nEnter the location at which to be deleted:”;***

***Cin >> loc;***

***If(head->next==NULL)***

***{***

***Ptr=head;***

***Head=NULL;***

***Free(ptr);***

***}***

***Else***

***{***

***Ptr=head;***

***For(int i=1;i<=loc;i++)***

***{***

***Ptr1=ptr;***

***Ptr=ptr->next;***

***If(ptr=NULL)***

***Return;***

***}***

***Ptr1->next=ptr->next;***

***Free(ptr);***

***}***

***}***

***Void deletion\_end()***

***{***

***Struct node \*ptr,\*ptr1;***

***If(head->next==NULL)***

***{***

***Ptr=head;***

***Head=NULL;***

***Free(ptr);***

***}***

***Else***

***{***

***Ptr=head;***

***While(ptr->next!=NULL)***

***{***

***Ptr1=ptr;***

***Ptr=ptr->next;***

***}***

***Ptr1->next=NULL;***

***Free(ptr);***

***}***

***}***

***Void display()***

***{***

***Struct node \*ptr;***

***Ptr=head;***

***If(head == NULL)***

***{***

***Cout << “\nNothing to print”;***

***}***

***Else***

***{***

***Cout << “\nThe values in list are: \n”;***

***While(ptr!=NULL)***

***{***

***Cout << ptr->data <<”\n”;***

***Ptr=ptr->next;***

***}***

***}***

***}***

***Void search()***

***{***

***Struct node \*ptr;***

***Int item,i=1;***

***Cout << “\nEnter an item to be searched:”;***

***Cin >> item;***

***Ptr=head;***

***While(ptr!=NULL)***

***{***

***If(ptr->data==item)***

***{***

***Cout << “\n” << item << “ is found at position: “ << I;***

***Break;***

***}***

***Else***

***Cout << “\n” << item << “doesn’t exist.”;***

***Ptr=ptr->next;***

***I++;***

***}***

***}***