```
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

typedef struct nodex
{
	int info;
	struct nodex* prev;
	struct nodex* next;
}node;

node* head = NULL;

void ins_b()
{
	int data;
	printf("Enter data: ");
	scanf("%d", &data);

	node* newNode = (node*) malloc(sizeof(node));
	newNode -> info = data;
	newNode -> next = newNode -> prev = NULL;
	if(head == NULL)
	{
		head = newNode;
		return;
	}
	else
	{
		newNode -> next = head;
		head -> prev = newNode;
		head = newNode;
	}
}


void ins_e()
{
	int data;
	printf("Enter data: ");
	scanf("%d", &data);
	node* newNode = (node *) malloc(sizeof(node));
	newNode -> info = data;
	newNode -> prev = newNode -> next = NULL;
	if(head == NULL)
	{
		head = newNode;
		return;
	}
	else
	{
		node* ptr;
		ptr = head;
		while(ptr -> next != NULL)
		{
			ptr = ptr -> next;
		}
		ptr -> next = newNode;
		newNode -> prev = ptr;
	}
```

```
}



void ins_pos()
{
      int i;
      int pos;
      int data;
      printf("Enter data: ");
      scanf("%d", &data);
      node* newNode = (node *) malloc(sizeof(node));
      newNode -> info = data;

      fputs("Enter position: ", stdout);
      scanf("%d", &pos);
      node* ptr;
      newNode -> prev = newNode -> next = NULL;
      for(i = 0; i < pos -1; ++i)
      {
            ptr = ptr -> next;
            if(ptr == NULL)
                  printf("\nNo nodes\n");
      }
      newNode -> next = ptr -> next;
      newNode -> prev = ptr;
      ptr -> next -> prev = newNode;
      ptr -> next = newNode;
}

void del_b()
{
      if(head == NULL)
      {
            printf("\nNo nodes!!!!\n\n");
            return;
      }
      node* ptr;
      ptr = head;

      head = head -> next;

      head -> prev = NULL;

      free(ptr);
}

void del_e()
{
      node* ptr;
      if(head == NULL)
      {
            printf("\nNo nodes!!!!\n\n");
            return;
      }
      if(head -> next == NULL)
      {
            ptr = head;

            head = NULL;
            printf("\nThe deleted item is: %d\n", ptr -> info);
            free(ptr);
      }
      else
      {
            ptr = head;
            while(ptr -> next != NULL)
            {
                  ptr = ptr -> next;
            }
            ptr -> prev -> next = NULL;
            printf("\nThe deleted item is: %d\n", ptr -> info);
            free(ptr);
      }
}


void del_pos()
{
      int pos;
      node* ptr;
      int i;
      fputs("Enter position: ", stdout);
      scanf("%d", &pos);
      ptr = head;
      if(head == NULL)
      {
            printf("\nNo nodes!!!!\n\n");
            return;
      }


      else
      {
            for(i = 0; i < pos; ++i)
            {
                  ptr = ptr -> next;
                  if(ptr == NULL)
                  {
                        printf("\nNo nodes!!!!\n\n");
                        return;
                  }
            }
            ptr -> prev -> next = ptr -> next;
            ptr -> next -> prev = ptr -> prev;

            free(ptr);
      }
}

void traverse()
{
      node* ptr;
      ptr = head;
      if(head == NULL)
      {
            printf("\nThe list is empty\n");
            return;
      }
      else
      {
            while(ptr != NULL)
            {

                          printf("  %d  ",ptr -> info);
                          ptr = ptr -> next;
            }
        printf("\n");
      }
}

int main(int argc,char **argv)
{
      int choice;
      while(true)
      {
            printf("\n1. Insert begining\n2.Insert Position\n3. Insert
End\n4. Delete Begining\n5.Delete Position\n6. Delete Pos\n7. Display\n8.
Exit\n");
            scanf("%d", &choice);
            switch(choice)
            {
                  case 1:
                        ins_b();
                        break;
                  case 2:
                        ins_pos();
                        break;
                  case 3:
                        ins_e();
                        break;
                  case 4:
                        del_b();
                        break;
                  case 5:
                        del_pos();
                        break;
                  case 6:
                        del_e();
                        break;
                  case 7:
                        traverse();
                        break;
                  case 8:
                        exit(0);
                        break;
                  default:
                        printf("\nEnter a better choice buddy!!!!\n");
            }
      }
      return 0;
}
```