Linked Lists

Insertion

```
#include <stdio.h>
#include <stdlib.h>
Struct node
{
   int data;
   Struct node *next;
};
void printData(struct node *head)
{
    if(head == NULL)
    {
       printf("The list is empty");
    }else{
    structi node *ptr=head;
    while(ptr!=NULL)
    {
       printf("%d\n", ptr->data);
       ptr=ptr->next;
    }}
 }
void insertBeg(struct node **head, int value)
{
   Struct node *temp=(struct node*)malloc(sizeof(struct node));
   temp->data=value;
   temp->next=*head;
   *head=temp;
}
void insertEnd(struct node *head, int value)
{
   Struct node *ptr=head;
   Struct node *temp=(struct node*)malloc(size of(struct node))
```

```
    temp -> data = value;
    temp -> next = NULL;
    while (ptr -> next != NULL) {
        ptr = ptr -> next;
    }
    ptr -> next = temp;

}
void insertAtPos (struct node *head, int value, int pos)
{
    struct node *ptr, *ptr2;
    struct node *temp = (struct node*)malloc (sizeof(struct node))
    temp -> data = value;
    temp -> next = NULL;
    int position = pos;
    ptr = head;
    while (pos != 1)
    {
        ptr2 = ptr;
        ptr = ptr -> next;
        pos--;
    }
    temp -> next = ptr2 -> next;
    ptr2 -> next = temp;
    printf("value %d added succesfully at %d\n", value, position)
```

```
void delBeg(struct node** head) {
      Struct node *ptr;
      if (head == NULL)
      {
          printf("The list is Empty");
      } else {
          ptr = *head;
          *head = (*head) -> next;
          free(ptr);
          ptr = NULL; }
}
void delEnd(struct node*head)
{
     Struct node *ptr, *ptr2;
     if (head == NULL) {
          printf("The List is Empty");
     } else {
       ptr = head;
       while (ptr->next = NULL) {
            ptr2 = ptr;
       ptr = ptr -> next; }
       ptr2 -> next = NULL;
       free(ptr);
}}
void delAtPos(struct node*head, int pos)
{
     struct node *ptr, *ptr2;
     if (head == NULL) {
          printf("The List is Empty\n");
}else if (pos==1)
{
     ptr=head;
     free(ptr);
     ptr=NULL;
}else{
ptr=head;
ptr2=head;
while(pos!=1){
    ptr2=ptr;
    ptr=ptr->next;
    pos--;
}
ptr2->next=ptr->next;
free(ptr);
ptr=NULL;
}}
int main()
{
  struct node *head=NULL;
  insertBeg(&head,34);
  printData(head);
  printf("-----\n");
  insertEnd(head,75);
  insertEnd(head,56);
  insertEnd(head,87);
  printData(head);
  printf("------\n");
  insertAtPos(head);
  ~~printData(head);~~ → insertAtPos(head, 89, 3);
  printf("___ ___ ___ - \n");
```

```
    delBeg (&head);
    printf ("-- -----\n");
    delEnd (head);
    printData (head);
    printf ("- - - - --\n");
    delAtPos (head, 2);
    printData (head);
    printf ("-- - - - --\n");
}
```

o/p:

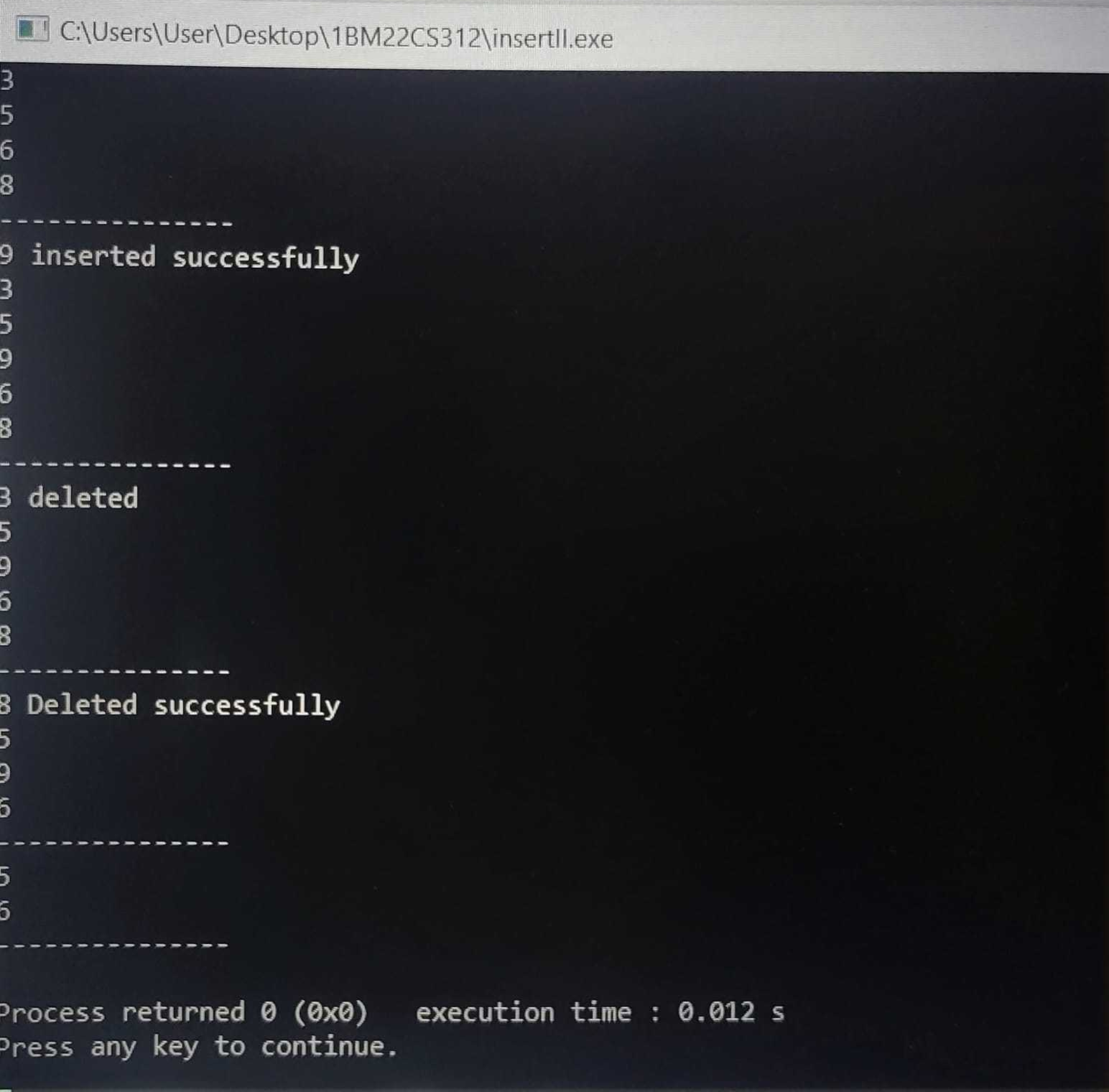
```
Value 34 added Succenfully at the Begining
34
-- - - - -
Value 75 added Successfully at the End
value 87 added succenfully at the End
value 54 added Succenfully at the End

34
75
87
54
value 89 added sucenfully at 3
34
75
89
87
54
--------
75
89
87
54
-- - - - -
```

```
75
89
87
------------
75
87
------------
```

✓

[illegible] 31/1/24


C:\Users\User\Desktop\1BM22CS312\insertll.exe

```
3
5
6
8
---------------
9 inserted successfully
3
5
9
6
8
---------------
3 deleted
5
9
6
8
---------------
8 Deleted successfully
5
9
6
---------------
5
6
---------------

Process returned 0 (0x0)   execution time : 0.012 s
Press any key to continue.
```