1.main.c

```
#include <stdio.h>
#include <stdlib.h>
#include "header.h"
#include <time.h>

int main() {
   srand(time(0));
   SLL L1;
   init_SLL(&L1);
   append(&L1);
   append(&L1);
   append(&L1);
   traverse(L1);
   insert_beg(&L1);
   printf("After inserting element at beginning: ");
   traverse(L1);
   remove_pos(&L1, 2);
   printf("After removing element from position 2: ");
   traverse(L1);
   len(L1);

  return 0;

}
```

2.header.h

```
typedef struct node{
   int data;
   struct node *next;
}node;

typedef node* SLL;

void init_SLL(SLL *head);
void append(SLL *head);
void traverse(SLL head);
void insert_beg(SLL *head);
void remove_pos(SLL *head, int pos);
void len(SLL head);
```

3.logic.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "header.h"
#include <time.h>
#define min 1
#define max 100
void init_SLL(SLL *head){
   *head = NULL;
   return;
}
void append(SLL *head){
   int randomNumber;
   int minimum = 1;
   int maximum = 100;
   randomNumber = (rand() % (maximum - minimum + 1)) + minimum;
   node *p, *newnode;
   newnode = (node*)malloc(sizeof(node));
   if(newnode){
      newnode->data = randomNumber;
      newnode->next = NULL;
   }
   else return;
   if(*head == NULL){
      *head = newnode;
      return;
   }
   p = *head;
   while(p -> next) {
     p = p->next;
```

```
    }
    p->next = newnode;
    return;
}
void traverse(SLL head){
    printf("[");
    node *p;
    p = head;
    while(p){
        printf("%d ", p->data);
        p = p->next;
    }
    printf("]\n");
    return;
}
void insert_beg(SLL *head){
    int randomNumber;
    randomNumber = (rand() % (max - min + 1)) + min;
    node  *newnode;
    newnode = (node*)malloc(sizeof(node));
    if(newnode){
        newnode->data = randomNumber;
        newnode->next = NULL;
    }
    else return;
    newnode->next = *head;
    *head = newnode;
    return;
}
void remove_pos(SLL *head, int pos){
    int i = 1;
```

```
    if (*head == NULL) {
        printf("The list is empty.\n");
        return;
    }
    node *temp = *head, *prev = NULL;
    if (pos == 1) {
        *head = temp->next;
        free(temp);
        return;
    }
    while (temp != NULL && i < pos) {
        prev = temp;
        temp = temp->next;
        i++;
    }
    if (temp == NULL) {
        printf("Position out of range.\n");
        return;
    }
    prev->next = temp->next;
    free(temp);
}
void len(SLL head) {
    int length = 0;
    node *temp = head;
    while (temp != NULL) {
        length++;
        temp = temp->next;
    }
    printf("The length of the list is: %d\n", length);
}
```

OUTPUT:

```
tanis@Tanishq MINGW64 /d/COEP/DSA/Assignments/8SinglyLinkedList
$ gcc -Wall main.c logic.c

tanis@Tanishq MINGW64 /d/COEP/DSA/Assignments/8SinglyLinkedList
$ ./a
[5 98 49 ]
After inserting element at beginning: [26 5 98 49 ]
After removing element from position 2: [26 98 49 ]
The length of the list is: 3
```