```
#include <stdio.h>
#include <stdlib.h>
/* Structure of a node */
struct node {
   int data;        // Data
   struct node *next; // Address
}*head;

void createList(int n);
void insertNodeAtEnd(int data);
void displayList();

int main()
{
   int n, data;

   /* Create a singly linked list of n nodes*/

   printf("Enter the total number of nodes: ");
   scanf("%d", &n);
   createList(n);

   printf("\nData in the list \n");
   displayList();
```

```
    /* Insert data at the end of the singly linked list */
    printf("\nEnter data to insert at end of the list: ");
    scanf("%d", &data);
    insertNodeAtEnd(data);

    printf("\nData in the list \n");
    displayList();

    return 0;
}

/* Create a list of n nodes */

void createList(int n)
{
    struct node *newNode, *temp;
    int data, i;

    head = (struct node *)malloc(sizeof(struct node));

    /* If unable to allocate memory for head node */
    if(head == NULL)
    {
        printf("Unable to allocate memory.");
    }
```

```
else
{
   /* Reads data of node from the user */
   printf("Enter the data of node 1: ");
   scanf("%d", &data);

   head->data = data; // Link the data field with data
   head->next = NULL; // Link the address field to NULL

   temp = head;

   /* Create n nodes and adds to linked list */
   for(i=2; i<=n; i++)
   {
      newNode = (struct node *)malloc(sizeof(struct node));

      /* If memory is not allocated for newNode */
      if(newNode == NULL)
      {
         printf("Unable to allocate memory.");
         break;
      }
      else
      {
         printf("Enter the data of node %d: ", i);
```

```
            scanf("%d", &data);

            newNode->data = data; // Link the data field of newNode with data
            newNode->next = NULL; // Link the address field of newNode with NULL

            temp->next = newNode; // Link previous node i.e. temp to the newNode
            temp = temp->next;
          }
        }

        printf("SINGLY LINKED LIST CREATED SUCCESSFULLY\n");
    }
}

/* Create a new node and inserts at the end of the linked list.*/

void insertNodeAtEnd(int data)
{
    struct node *newNode, *temp;

    newNode = (struct node*)malloc(sizeof(struct node));

    if(newNode == NULL)
    {
        printf("Unable to allocate memory.");
```

```
    }
    else
    {
        newNode->data = data; // Link the data part
        newNode->next = NULL;

        temp = head;

        // Traverse to the last node
        while(temp != NULL && temp->next != NULL)
            temp = temp->next;

        temp->next = newNode; // Link address part

        printf("DATA INSERTED SUCCESSFULLY\n");
    }
}



/*Display entire list */

void displayList()
{
    struct node *temp;
```

```
    /* If the list is empty i.e. head = NULL */
    if(head == NULL)
    {
        printf("List is empty.");
    }
    else
    {
        temp = head;
        while(temp != NULL)
        {
            printf("Data = %d\n", temp->data); // Print data of current node
            temp = temp->next;              // Move to next node
        }
    }
}
```