1. main.c

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

int main() {
    ASCII L1;
    init_ASCII(&L1);
    ASCII_of(&L1);
    traverse(&L1);
    Destroy(&L1);
    traverse(&L1);
    return 0;
}
```

2. header.h

```c
typedef struct node{
    int data;
    struct node *prev, *next;
}node;

typedef struct ASCII{
    node *front;
    node *rear;
}ASCII;

void init_ASCII(ASCII *l);
int isEmpty(ASCII l);
void ASCII_of(ASCII *l);
void traverse(ASCII *l);
void Destroy(ASCII *l);
```

3. logic.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "header.h"
void init_ASCII(ASCII *l){
    l -> front = NULL;
    l -> rear = NULL;
    return;
}
int isEmpty(ASCII l){
    if(l.front == NULL){
        return 1;
    }
    else{
        return 0;
    }
}
void ASCII_of(ASCII *l){
    node *nn;
    char ch;
    int value, lastdigit;
    printf("Enter the character: ");
    scanf("%c", &ch);
    value = (int)ch;
    while(value){
        lastdigit = value % 10;
        nn = (node *)malloc(sizeof(node));
        if(nn){
            nn -> data = lastdigit;
            nn -> next = NULL;
            nn -> prev = NULL;
```

```c
        }
        else{
            return;
        }
        if(!isEmpty(*l)){
            node *p = l -> front;
            p -> prev = nn;
            nn -> next = p;
            l -> front = nn;
        }
        else{
            l -> front = nn;
            l -> rear = nn;
        }
        value = value / 10;
    }
}

void traverse(ASCII *l){
    node *p = l -> front;
    if(p == NULL){
        printf("The list is empty\n");
        return;
    }
    while(p != NULL){
        printf("%d ", p -> data);
        p = p -> next;
    }
    printf("\n");
}
```

```
void Destroy(ASCII *l){

    node *p = l -> front;

    node *temp;

    while(p != NULL){

        temp = p;

        p = p -> next;

        free(temp);

    }

    l -> front = NULL;

    l -> rear = NULL;

    printf("Destroying of Linked List is completed\n");

}
```

 4. Output of the code