```
#include<stdio.h>
#include<stdlib.h>
#include "Linked_List.h"
#define _CRT_SECURE_NO_WARNINGS

int main() {
    struct Linked_List* start = NULL;
    int arr[100] , count = 0 , temp , number = 0;
    printf("Enter The numbers for the linked list.\nEnter -1 for ending the list. \nThe list: ");
    for (int i = 0; i < 100; i++){
        scanf_s("%d", &arr[i]);
        if (arr[i] != -1) {
            Build_Linked_List(&start, arr[i]);
            count++;
        }
        else
            break;
    }
    Print_Linked_List(start);
    printf("\n\nEnter number to the linked list and the funcation will organization in the right order all numbers: ");
    scanf_s("%d", &arr[count+1]);
    Build_Linked_List(&start, arr[count+1]);
    Sort_Linked_List(start, count + 1);
    Print_Linked_List(start);
    printf("\n\nEnter a number and the funcation will find if the exists: ");
    scanf_s("%d", &number);
    temp = Search_Number(start, number);
    if (temp>0)
        printf("\nindex in list: %d", temp);
    else
        printf("\n%d dosent exists in the list.", number);
    Reverse_Linked_List(&start);
    printf("\n\nReversed Linked list:");
    Print_Linked_List(start);
    temp = 0;
    Sum_Linked_List(start, &temp);
    printf("\n\nSum of all numbers in the list: %d", temp);
    temp = 1;
    Multiplication_Even_Indexs_Linked_List(start, &temp, 0);
    printf("\n\nMultiplication all numbers that are in the even places on the list: %d\n", temp);
    return 0;
}
```

```
#pragma once
#include<stdio.h>
#include<stdlib.h>
#define _CRT_SECURE_NO_WARNINGS

struct Linked_List{
    int Data;
    struct Linked_List* Head_To_Tail;
};

void Build_Linked_List(struct Linked_List**,int);
void Sort_Linked_List(struct Linked_List*,int);
void Swap_Numbers(struct Linked_List*,struct Linked_List*);
int Search_Number(struct Linked_List*,int);
void Reverse_Linked_List(struct Linked_List**);
void Sum_Linked_List(struct Linked_List*,int*);
void Multiplication_Even_Indexs_Linked_List(struct Linked_List*,int*,int);
void Print_Linked_List(struct Linked_List*);
```

```
#include "Linked_List.h"

void Build_Linked_List(struct Linked_List** Building, int Move_To_Data){
    struct Linked_List* Temp_Struct = (struct Linked_List*)malloc(sizeof(struct    ↩
      Linked_List));
    Temp_Struct->Data = Move_To_Data;
    Temp_Struct->Head_To_Tail = *Building;
    *Building = Temp_Struct;
}

void Sort_Linked_List(struct Linked_List* Start_To_End,int count) {
    int Swap = 0;
    struct Linked_List* Temp_1;
    struct Linked_List* Temp_2 = NULL;
    for (int i = 0; i < count; i++){
        Swap = 0;
        Temp_1 = Start_To_End;
        while (Temp_1->Head_To_Tail != Temp_2) {
            if (Temp_1->Data > Temp_1->Head_To_Tail->Data) {
                Swap_Numbers(Temp_1, Temp_1->Head_To_Tail);
                Swap = 1;
            }
            Temp_1 = Temp_1->Head_To_Tail;
        }
        Temp_2 = Temp_1;
    }
}

void Swap_Numbers(struct Linked_List* Number_1, struct Linked_List* Number_2){
    int temp = Number_1->Data;
    Number_1->Data = Number_2->Data;
    Number_2->Data = temp;
}

int Search_Number(struct Linked_List* From_Start_To_End, int Index){
    struct Linked_List* current = From_Start_To_End;
    int count = 0;
    while (current != NULL){
        count++;
        if (current->Data == Index)
            return count;
        current = current->Head_To_Tail;
    }
    return 0;
}

void Reverse_Linked_List(struct Linked_List** From_Start_To_End){
    struct Linked_List* Right_Order = *From_Start_To_End;
    struct Linked_List* Bad_Order = NULL;
    struct Linked_List* Right_Order_Next = NULL;
    while (Right_Order != NULL) {
        Right_Order_Next = Right_Order->Head_To_Tail;
        Right_Order->Head_To_Tail = Bad_Order;
        Bad_Order = Right_Order;
        Right_Order = Right_Order_Next;
    }
    *From_Start_To_End = Bad_Order;
}

void Sum_Linked_List(struct Linked_List* From_Start_To_End, int* Sum) {
    if (!From_Start_To_End)
        return;
    Sum_Linked_List(From_Start_To_End->Head_To_Tail, Sum);
    *Sum = *Sum + From_Start_To_End->Data;
}

void Multiplication_Even_Indexs_Linked_List(struct Linked_List* From_Start_To_End, ↩
  int* Sum, int Index) {
    if (!From_Start_To_End)
        return;
    Index++;
    if (Index % 2 == 0)
        *Sum = *Sum * From_Start_To_End->Data;
    Multiplication_Even_Indexs_Linked_List(From_Start_To_End->Head_To_Tail, Sum,   ↩
      Index);
}

void Print_Linked_List(struct Linked_List* From_Start_To_End) {
    struct Linked_List* temp = From_Start_To_End;
    int count = 0;
    printf("\nThe List is: ");
    while (temp != NULL)
    {
        printf("%d ", temp->Data);
        temp = temp->Head_To_Tail;
        count++;
    }
    printf("\nThe Lenth is: %d", count);
}
```