Program:

```
# include <stdlib.h>
# include <stdio.h>
# include <malloc.h>
 struct node {
      int data;
      struct node * next;
  };
   struct node * start = NULL, *start1 = NULL;
   struct node * create_ll(struct node *);
   struct node * create_l2(struct node *);
   struct node * display(struct node *);
   struct node * reverse(struct node *);
   struct node * concat(struct node *);
   struct node * sort_list(struct node *);
    int main() {
         int option;
         do {
           printf("\n1. Create a list");
           printf("\n2. Display a list");
           printf("\n3. Reverse entire list");
            printf("\n4. Concatenation of two list");
            printf("\n5. Sort the list");
            printf("\n6. EXIT");
            printf("\nEnter your choice : ");
            scanf("%d", &option);
             switch(option) {
             case 1: start = create_ll(start);
               printf("\n Linked list created");
                 break;
case 2 : start = display (start) ;
   break;
 ~~case 3: start = insert_beg (start);~~
  ~~break;~~
 ~~case 4: start = insert_end (start);~~
  ~~break;~~
 ~~case 5: start = insert_~~
  case 3 : start = reverse (start) ;
   break;
  case 4 : start = concat (start) ;
   break;
  case 5: start = sort_list (start);
   break;
 }
} while (option != 6) ;
 return 0;
}
struct node * create_ll ( struct node* start ) {
  struct node * ptr, * new_node ;
  int num;
  printf ("\n Enter -1 to end");
  printf (" Enter the data");
  scanf ("%d",& num);
  while ( num != -1) {
     new_node = (struct node *) malloc (sizeof(struct
                                            node));
   new_node -> data = num;
    if ( start == NULL) {
       new_node -> next = NULL;
       start = new_node;
    }
```

```
else {
     ptr = start;
    while (ptr->next ! = NULL) {
          ptr = ptr->next;
          ptr->next = new_node;
          new_node->next = NULL;
    }
     printf("\n Enter the data");
     scanf("%d", &num);
   }
   return start;
 }
~~struct node * display(struct node* start){~~
    ~~struct node* new_node;~~
    ~~int num;~~
    ~~printf("Enter the data");~~
    ~~scanf("%d", &num~~
struct node * display (struct node* start){.
    struct node * ptr;
     ptr = start;
    while (ptr! = NULL) {
       printf("\t %d", ptr->data);
        ptr = ptr->next;
     }
     return start;
  }.
struct node * reverse (struct node * start) {
 struct node * prev = NULL, * current = start,
          * next = NULL;
       while(current! = NULL) {
            next = current->next;
            current -> next = prev;
            prev = current;
            current = next;
        }
        start = prev;
        return start;
    }
struct node* concat(struct node* start, struct node* start1)
{   struct node *ptr;
    ptr = start;
    printf("Enter the second");
    start1 = create_l2(start1);
    while (ptr -> next = NULL){
        ptr = ptr -> next;
    ptr -> next = start1;
    return start;
}
struct node * sort_list(struct node * start){
    struct node * ptr1, *ptr2;
    int temp;
    ptr1 = start;
    while (ptr1 -> next != NULL){
        ptr2 = ptr1 -> next;
        while(ptr2 != NULL){
            if (ptr1 -> data > ptr2 -> data){
                temp = ptr1 -> data;
                ptr1 -> data = ptr2 data;
                ptr2 -> data = temp;
            }
            ptr2 = ptr2 -> next;
        return start;
    }
```