## Program 6a

```
Write A Program to Implement Single Link List with following operations:
Sort the linked list,
Reverse the linked list,
Concatenation of two linked lists.
Code:
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
  int data;
  struct Node* next;
} Node;
Node* createNode(int data) {
  Node* newNode = (Node*)malloc(sizeof(Node));
  if (!newNode) {
    printf("Memory error\n");
    return NULL;
  newNode->data = data;
  newNode->next = NULL;
  return newNode;
}
void insertNode(Node** head, int data) {
  Node* newNode = createNode(data);
  if (*head == NULL) {
    *head = newNode;
    return;
  Node* lastNode = *head;
  while (lastNode->next) {
    lastNode = lastNode->next;
  lastNode->next = newNode;
}
void printlist(Node* head) {
  Node* current = head;
  while (current) {
    printf("%d -> ", current->data);
    current = current->next;
  printf("NULL\n");
```

}

```
void sortlist(Node* head) {
  if (head == NULL) {
    return;
  Node* current;
  Node* nextNode;
  int temp;
  for (current = head; current != NULL; current = current->next) {
    for (nextNode = current->next; nextNode != NULL; nextNode = nextNode->next) {
       if (current->data > nextNode->data) {
         temp = current->data;
         current->data = nextNode->data;
         nextNode->data = temp;
  }
}
void reverselist(Node** head) {
  Node* prev = NULL;
  Node* current = *head;
  Node* next = NULL;
  while (current) {
    next = current->next;
    current->next = prev;
    prev = current;
    current = next;
  *head = prev;
void concatenate(Node** head1, Node* head2) {
  if (*head1 == NULL) {
     *head1 = head2;
    return;
  Node* lastNode = *head1;
  while (lastNode->next) {
    lastNode = lastNode->next;
  lastNode->next = head2;
}
int main() {
  Node* head1 = NULL;
  Node* head2 = NULL;
  insertNode(&head1, 1);
  insertNode(&head1, 3);
  insertNode(&head1, 5);
```

```
insertNode(&head2, 2);
  insertNode(&head2, 4);
  insertNode(&head2, 6);
  printf("Linked list 1: ");
  printlist(head1);
  printf("Linked list 2: ");
  printlist(head2);
  sortlist(head1);
  printf("Sorted Linked list 1: ");
  printlist(head1);
  reverselist(&head2);
  printf("Reversed Linked list 2: ");
  printlist(head2);
  concatenate(&head1, head2);
  printf("Concatenated Linked list: ");
  printlist(head1);
  return 0;
}
```

```
Linked list 1: 1 -> 3 -> 5 -> NULL
Linked list 2: 2 -> 4 -> 6 -> NULL
Sorted Linked list 1: 1 -> 3 -> 5 -> NULL
Reversed Linked list 2: 6 -> 4 -> 2 -> NULL
Concatenated Linked list: 1 -> 3 -> 5 -> 6 -> 4 -> 2 -> NULL
```

Node \* newhode = \* head;

Node \* Newhole = \* head;

? (boad \*\* shan) kiloweyare his while (Inithlade -> next)? Hade \* paren = Mulling \*\* head)
Nade \* considert \* \* head,
Nade \* most = NIKI;
while (current) \* 3 (adheric - rext = rewhade; next = Custient > next = press,

Perext = Custient > next = press,

Perext = Custient;

Questient = next; void pointful (Node \* head ) ? points (' d \rightarrow ', head \rightarrow data);
head = head \rightarrow next; \* head = paies; point] (" NULL 'n"); void southint (Node \* Head) { used commitments (nede \*\* head);

used (head) = Nede \* head 2);

\* head = head 2;

steem; Ned \* Cusuent = head;

If thend = NULL)?

Authorn; ushile (cussent) & Nede \* hutNede = \* head 1; Whale (lastNede - next) 9; JastNede = lastNede - next; Made \* northade = convent > northade = doing! temp = cusuant -> data;
current -> data = neathble -> data;
revinade -> data = temp; luthede -> next = heada. rextNede = nextNedo -> next; Nede \* head 1 = NULL;
Nede \* head 2 = NULL;
Nede \* head 3 = NULL;
Nede \* head 1, 1);
(Shead 3);
(Shead 5); cusyomt = cusyomt -> rext;

investible (pheade, 2); (gheada, 4); paint (" Linked List 1: "); pounts (" united lat 2: "). pointlist (head a). contlist (head 1); of (" Serted Linked dist 1: ") countlut (head 1);
several tit (dhead 2);
of ("Reversed Linked List 2:"); pointlet (head &); concatenate (shead 1, head 2); of ( Concatenated linked list = 2 eneturn O; output bused July 1:  $1 \rightarrow 3 \rightarrow 5 \rightarrow \text{NULL}$ Virked July 2:  $2 \rightarrow 4 \rightarrow 6 \rightarrow \text{NULL}$ ested linked list  $1:1\rightarrow3\rightarrow5\rightarrow$  KNUL excreted linked list  $2:6\rightarrow4\rightarrow2\rightarrow$  NULL concated linked list:  $1 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 4 \rightarrow 2 \rightarrow$