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
```

	and the second s
store 67	- 6
	And Company And
who to implement a Simply linked like whe is	while (last Node -> nert) f
up to implement a Singly linked list with the fall	last Node = last Node > next;
	The California Carlotte Control of the Carlotte Contro
Sort the linked eigt	lastNode -> next = newNode;
Reverse the linked list	
Concatinating 2 linked lists.	void partlist(Node + head)
0	
#include < Atdio. N)	mist (" , dia", head - data/
Lincude (cidliph)	hend = head -> nest;
Struck Node 9	3
int data: (Continue) 3 mod	minel (Noul In");
Struct noole + mentioners;	prince (NULL IN)
3 Node;	Void Sortlist (Node * hend) {
Salah Wasan Salah	Void Sortist incor news
Node+ create Note (int data) {	Note + current > head;
Node* new Node . (Node*) mallo ((size of (Node));	If (head = NVLL)
if (! new Node) {	Yelarn,
	YCHAYN,
Printf ("Memory ener In"); Yellurn NVIL;	3
Yelam NVIC,	while (current) {
They to brown and they high	Node+ next Node = current → next;
nawNode > data = data	while (next Node)f
new Node - next = NULL	if Gunerat Fidata > next Node 7
return newNode;	Jemp=cumut >data;
Charles Alab & Harmin	current 7 days = next Node >
void inserthode (Node ** head, int data) f	neravide - dala= kmp;
Node * new Node = (rioteNode (data),	Callabora Republication
and white and if thread = 2 Nort) f	neithede = next Node -> nex
+ head = new Node;	Method > Marion
Yelum	Company and a superior
	3 Carrent = Carrent -> Next;
Node+ last Node = + hend;	36 and the shared and the shared
NOCE ABSTROLE TON	and sol that
67	and the same of th
	19
Void reverselist (Node ++ nega) {	
Node* prev = NULL;	inert Node (lhord 2, 2);
Node* current = + head;	Chread 2, 42;
Nide + next = NULL;	(haar 2, 6)
While Carrent)?	prive ("Linked list 1:")
next : Current - next;	mintist (head)
Current > next > prev;	prints Cheed 1/2:");
prev = carrent;	print Clinted let 2:
Carlos Carlos	Divitlist head the

Void revenelist (Node ** news) {
Node* prev: Noll;
Node* Current ** head;
Node* next = Noll;
while (Current) {
next = Current = rest;
Current > next;
Current > next;

Current > next;

grev = current;

Current | next;

}

* head: prev;

}

void concuentate (Node ** head), Node* head 2) {
if (head != Nov!) }

* head ! head 2;

Yellury;

}

Node* Latinode = Aread;
while (latinode > next;

last Node = latinode > next;

int main()

Node* head != NULL;

Vode * head != NULL;

Vode * head != NULL;

insert Node (Aread), 1);
(Shoul ; 3);
(Encot), 5);

inart Node (sheed 2 , 2);

(sheed 2 , 4);

(sheed 2 , 4);

(sheed 2 , 4);

print ("Linked & 1 = ");

print (sheed 2);

Sort List (head 2);

Sort List (head 2);

print (sheed 1);

print (sheed 2);

print (sheed 2