7(6) Singly linked list (sort, concatenate, reverse, display) program

```
#include <stdio.h>
#include <stdlib.h>
void sort ();
void create1 () ;
void reverse ();
void create2 () ;
void concatenate () ;
void display ();
struct node {
int data ;
struct node *next ;
};
struct node *head = NULL ;
struct node *head2 = NULL ;
int c ;
int main () {
int choice ;
do {
printf("\n1.Create \n2.Sort Linked List \n3 Reverse Linked List
\n4.Concatenate two linked lists \n5.Display \n6. EXIT");
printf ("Enter your choice : ") ;
scanf ("%d", &choice );
Switch (choice) {
case 1 : create1 () ; break;
case 2 : sort () ; break ;
case 3 : reverse () ; break;
case 4: create2 () ; ~~break;~~
        concatenate () ; break;
case 5 : display () ; break;
case 6 : printf ("\n <- - - - - Exiting the program - - - - - -> \n"); break;
}
}while (choice != 6);
return 0;
}
void create1 () {
struct node *newnode;
struct ~~node~~ node *temp;
int s;
printf ("Enter integer :");
scanf ("%d", &s);
newnode = (struct node*) malloc (size of (struct node));
newnode -> data = s;
if (head == NULL) {
newnode -> next = NULL;
head = newnode;
printf ("First node of linked list created \n");
c++;
}else {
temp = head
while (temp -> next != NULL)
temp = temp -> next;
temp -> next = newnode;
newnode -> next = NULL;
c++;
printf ("NODE created \n");
}
}
void reverse (){
struct node *prev = NULL, *current = head, *next = NULL;
while (current != NULL) {
next = current -> next;
```

```
current → next = prev ;
prev = current ;
current = next ;
}
head = prev ;
print ("The list is reversed successfully !! \n") ;
}
void display () {
struct node *ptr = NULL ;
ptr = head ;
if (ptr = = NULL)
printf ("Nothing to print \n") ;
else {
printf ("\n Contents of the Linked List \n") ;
while (ptr ! = NULL) {
 printf ("%d \t ", ptr → data) ;
  ptr = ptr → next ;
}
}
printf ("\n") ;
}
void create2 () {
struct node *newnode ;
struct node *temp ;
int s, y ;
printf ("Enter elements to the SECOND Linked List 2 \n") ;
do {
printf ("Enter integer : \n") ;
scanf ("%d ", &s) ;
newnode = (struct node*) malloc (size of (struct node) ) ;
newnode → data = s ;
if (head2 = = NULL) {
```

```
newnode -> next = NULL ;
head2 = newnode ;
printf ("First node of linked list created \n");
c++ ;
} else {
temp = head2 ;
while (temp -> next != NULL)
temp ~~->~~ = temp -> next ;
temp -> next = newnode ;
newnode ~~of~~ -> next = NULL ;
c++ ;
printf ("Node created \n");
}
printf ("Do you want to continue adding : \n 0 = NO or 1 = Yes \n");
scanf ("%d, &y) ;
} while(y != 0) ;
}

void concatenate () {
 struct node *ptr ;
 if (head == NULL)
  head = head2;
if (head2 = NULL)
 head2 = head ;
 ptr = head ;
while (ptr -> next != NULL)
 ptr = ptr -> next ;
ptr -> next = head2 ;
}

void sort () {
 int swap, i ;
 struct node *ptr1 ;
struct node *lptr = NULL ;
if (head == NULL)
return ;
do {
swap = 0 ;
 ptr1 = head ;
while (ptr1 -> next != lptr ){
if (ptr1 -> data > ptr1 -> next -> data )
{
 int temp  = ptr1 -> data;
ptr1 -> data = ptr1 -> next -> data;
 ptr1 -> next -> data = temp;
swap = 1;
}
ptr = ptr1 -> next ;
}
lptr = ptr1 ;
}
While (swap) ;
}
```