Date
Page

LAB 7

```
# include <stdio.h>
# include <stdlib.h>
void sort();
void create ();
void reverse ();
void create_second();
void ~~coate cond~~ 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(" 1. Create\n 2. Sort\n 3. Reverse\n 4. Enter second lis
                5. Concatenate\n 6. Display \n 7. Exit \n Enter choice"
            scanf("%d", &choice);
            switch(choice){
                case 1: create(); break;
                case 2: sort(); break;
                case 3: reverse (); break;
                case 4: create_second(); break;
                case 5: concatenate(); break;
                case 6: display(); break;
            }~~case~~
        } while (ch!= 7);
}
```

```
void create(){
    struct node * newnode~~*~~, * temp;
    int s;
    printf("Enter data");
    scanf("%d", &s);
    newnode = (struct node *) malloc(size of (struct node));
    newnode -> data = s;
    if(head == NULL){
        newnode -> next = NULL;
        head = newnode;
        printf("Created");
        c ++;
    }
    else{
        temp = head;
        while( temp->next != NULL)
            temp = temp -> next;
        temp -> next = newnode;
        newnode -> next = NULL;
        c ++;
    }
}
void reverse(){
    struct node *prev = NULL, * current = head, * next = NULL;
    while (current != NULL){
        next = current ->next;
        current -> ~~prev~~ next = prev;
        prev = current;
        current = next;
    }
    head = prev;
}
```

```
void display(){
    struct node *ptr = NULL;
    ptr= head;
    if(ptr== NULL)
        printf("Empty list");
    else{
        while(ptr != NULL){
        {   printf("%d\t", ptr->data);
            ptr = ptr->next;
        }
    }
}
void create_second(){
    struct node * newnode, *temp;
    int ~~x~~ y;
    ~~ptr~~ newnode = (struct node *) malloc(size of(struct node));
    printf("Enter data:");
    scanf("%d", newnode->data);

    if(head 2 == NULL){
        newnode->next = NULL;
        head2 = newnode;
        c++;
    }
    else{
        temp = head2;
        while(temp->next != NULL)
            temp= temp->next;
        temp->next = newnode;
        newnode->next = NULL;
        c++;
    }
```

```
void  concatenate (){
        struct  node  *ptr;
        if(head == NULL)
              head = head 2;
         if( head2 = = NULL)
              head 2 = head;
         ptr = head;
         while (ptr->next != NULL)
                ptr = ptr -> next;
         ptr -> next = head 2;
}
void  sort () {
        int  swap, i;
        struct  node  *ptr 1, *1ptr =NULL;
        if (head == NULL)
             return;
        do{
             swap = 0;
             ptr1 = head;
             while (ptr 1->next != 1ptr) {
                  if (ptr 1->data > ptr 1 ->next ->data){
                        int  temp = ptr1->data;
                        ptr 1 -> data = ptr1 ->next->data;
                        ptr1 -> next -> data = temp;
                        swap=1;
                  }
                  ptr1= ptr1 -> next;
             }
             1 ptr= ptr1;
        } while (swap);
  }
```