Date / /
Page

WEEK 6 :-

1) Singly linked list :- sort, reverse & concatenation

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

typedef struct node {
        int data
        struct node* next ;
} node;

node *head = NULL;
node *head =NULL;
int count =0;

void insert (int data, int position );
void delete (int position );
void display ();
void sort ();
void reverse ();
void contact (node** head1, node** head2);

int main ()
{
  int data, choice, pos;
  printf ("1. Insert \n 2. Delete  3. Exit \n choice :");
  scanf ("%d", &choice);
  while (choice != 3)
  {
      if (choice == 1);
      {
```

```
printf ("Enter data and position : ");
scanf ("%d %d", &data, &pos);
insert (data, pos);
printf (" Count : %d\n", count);
}
else if (choice == 2)
{
printf ("Enter position :")
scanf ("%d", &pos);
delete (pos);
printf ("Count : %d\n ", count);
}
display ();
printf ("Enter choice :");
scanf ("%d", &choice);
}
printf ("original linked list :\n");
display ();
sort ();
printf ("sorted linked list :\n");
display ();
reverse ();
printf ("Reverse linked list :\n");
display ();
head1 = head;
head = NULL;
insert (3,0);
insert (4,1);
insert (1,2);
display ();
concat (&head1, &head);
head = head1;
```

Date __/__/__
Page

```
printf ("concatenation with the above linked list
        given: \n");
return 0;
}
void sort ()
{
    int i, j, min_index;
    node * i_node = head, * j_node = head, * min_node = NULL;
    for (int i = 0; i < count - 1; i++; i_node = i_node->
                                        next).
    {
        for (j_node -> data < i_node -> data).
        {
            min_index = j;
            min_node = j_node;
        }
    }
    if (min_index != i).
    {
        int temp = i_node -> data;
        i_node -> data = temp;
    }
}
}
void reverse ()
{
    node * prev = NULL, * next = NULL;
    while (head != NULL).
    {
        next = head -> next
        head -> next = prev
        prev = head;
```

```
    head = next; }
head = prev; }
void concat (node **head1, node **head2)
{
    node *temp1 = *head2;
    while (temp1 -> next != NULL)
    {
        temp1 = temp1 -> next;
    }
    temp1 -> next = *head2;
}}
```

OUTPUT:-

Original linked list:
2 1 4 3 5

Sorted linked list:
1 2 3 4 5

Reversed linked list:
5 4 3 2 1
3 4 1

Concatination with the above linked list give:
5 4 3 2 1 3 4 1

Date / /
Page

STACK USING LINKED LIST:-

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

typedef struct Node {
        int data;
        struct Node * next;
} node;

node * head = NULL;
int count = 0;

void insert (int data);
int delete ();
void display ();

int main ():
{
    int data, choice, pos;
    printf ("1. Insert \n 2. Delete \n 3. Exit \n Choice:")
    scanf ("%d", &choice);
    while (choice != 3)
    {
        if (choice = 1)
        {
            printf ("Enter data :");
            scanf ("%d", &data);
            insert (data);
            printf ("Count : %d \n", count);
}
```

Date __/__/__
Page _____

```
else if (choice == 2)
{
    printf ("Integer popped = %d\n", delete());
    printf ("Count : %d\n", count);
}
display();
printf ("Enter choice : ");
scanf ("%d", &choice);
}
return 0;
}
void insert (int data)
{
    node * new_node = (node*) malloc (sizeof (node));
    new_node -> data = data;
    new_node -> next = head;
    head = new_node;
    count ++;
    return;
}
void display ()
{
    node * temp = head;
    printf ("Stack : ");
    while (temp -> next != NULL)
    {
        printf ("%d", temp -> data);
        temp = temp -> next;
    }
    printf ("%d", temp -> data);
    printf ("\n");
}
```

11/12/24

Date __/__/__
Page _____

OUTPUT:-

1: Insert
2: Delete
3: Exit
Choice: 1
Enter data: 1
Count: 1
stack: 1
Enter choice: 1
Enter data: 3
Count: 2
stack: 3 1
Enter choice 2
Integer popped = 3
Count: 1
stack: 1
Enter choice: 3