

- i) Write code to implement Singly Linked List with following operations
 - a) Create linked list
 - b) Insertion of a node at first position, at any position and End of list.

Display the contents of the linked list

```
#include <stdio.h>
#include <malloc.h>

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

struct node * start = NULL;
struct node * create (struct node * );
struct node * display (struct node * );
struct node * insert_beg (struct node * );
struct node * insert_end (struct node * );
struct node * insert_before (struct node * );
struct node * insert_after (struct node * );
struct node * sort_list (struct node * );
struct node * sort_dlist (struct node * );

int main ()
{
    int choice;
    printf ("\n\n + menu ++ \n1. Create va list \t2. dis
    \t3. insert-beg \t4. insert end \t5. insert before
    insert after \t6. sort \n");
}
```

```

do
{
    printf ("In enter the choice : ");
    scanf ("%d", &choice);
    switch (choice)
    {
        case 1 : start = create_ll (start);
                    printf ("In linked list created");
                    break;
        case 2 : start = display (start);
                    break;
        case 3 : start = insert_beg (start);
                    break;
        case 4 : start = insert_end (start);
                    break;
        case 5 : start = insert_before (start);
                    break;
        case 6 : start = insert_after (start);
                    break;
        case 7 : start = sort_list (start);
                    break;
    }
}

while (choice != 8);
return 0;
}

```

```

struct node * create_ll (struct node * start)
{
    struct node * new_node, * ptr;
    int num;
    printf ("In enter -1 to end ");
    printf ("In enter the data : ");

```

```

scanf("%d", &num);
while (num != 1)
{
    new_node = (struct node*) malloc (sizeof (struct
    new_node->data.num;
    if (start == NULL)
    {
        new_node->next = NULL;
        start = new_node;
    }
    else
    {
        ptr = start;
        while (ptr->next != NULL)
            ptr = ptr->next;
        ptr->next = new_node;
        new_node->next = NULL;
    }
    printf("Enter the data : ");
    scanf("%d", &num);
}
return start;
}

```

```

struct node *display (struct node *start)
{
    struct node *ptr;
    ptr = start;
    while (ptr != NULL)
        printf("%d", ptr->data);
    ptr = ptr->next;
}

```

```
return start;  
};  
struct node * insert_beg (struct node * start)  
{  
    struct node * new_node;  
    int num;  
    printf ("\n enter the data :");  
    scanf ("%d", &num);  
    new_node = (struct node *) malloc (size of  
        node);  
    new_node->data = num;  
    new_node->next = start;  
    start = new_node;  
    return start;  
};
```

```
struct node * insert_end (struct node * start)  
{  
    struct node * new_node, * p;  
    int num;  
    printf ("\n enter the data :");  
    scanf ("%d", &num);  
    new_node = (struct node *) malloc (size of  
        node);  
    new_node->data = num;  
    new_node->next = start;  
    start = new_node;  
    return start;  
};
```

struct node * insert_before (struct node * start)

```
struct node * new-node, * ptr, * preptr;  
int num, val;  
printf("In enter the data :");  
scanf("%d", &num);  
printf("In enter the value before which data has to  
be inserted :");  
scanf("%d", &val);  
new-node = (struct node *) malloc(sizeof(struct node));  
new-node->data = num;  
ptr = start;  
while (ptr->data != val)  
{  
    preptr = ptr;  
    ptr = ptr->next;  
}  
preptr->next = new-node;  
new-node->next = ptr;
```

return start;

b:

struct node * insert_after (struct node * start)

a

```
struct node * new-node, * ptr, * preptr;  
int num, val;  
printf("In enter the data :");  
scanf("%d", &num);  
printf("In enter the value after which data has to  
be inserted :");  
scanf("%d", &val);
```

```
new_node = (struct node *) malloc (size of struct node)
new_node->data = num;
ptr = start;
while (ptr->data != val)
{
    preptr = ptr;
    ptr = ptr->next;
}
preptr->next = new_node;
new_node->next = ptr;
return start;
```

```
struct node *sort_list (struct node *start)
{
    struct node *ptr1, *ptr2;
    int temp;
    ptr1 = start;
    while (ptr1->next != NULL)
    {
        ptr2 = ptr1->next;
        while (ptr2->next != NULL)
        {
            if (ptr1->data > ptr2->data)
            {
                temp = ptr1->data;
                ptr1->data = ptr2->data;
                ptr2->data = temp;
            }
            ptr2 = ptr2->next;
        }
        ptr1 = ptr1->next;
    }
}
```

ptr1 = ptr1 -> next;

}
display (start);
};

Output:

1. break linked list
2. display.
3. insert - beg.
4. insert - end.
5. insert - before.
6. insert - after

linked list create enter the choice :

9.

Enter the num : 10

Enter the num : 20

Enter the num : 30

Enter the num : -1.

Enter your choice : 2.

10 20 30

Enter your choice : 3.

Enter the num : 15

Enter your choice : 2.

15 10 20 30.

2) WAP to implement singly linked list with following operations.

- Create a linked list.
- Deletion of first element, specified element and
element in the list.
- Display the contents of the linked list

```
include <stdio.h>
#include <malloc.h>
struct node
{
    int data;
    struct node *next;
};

struct node *start = NULL;
struct node *create_ll (struct node * );
struct node *display (struct node * );
struct node *delete_beg (struct node * );
struct node *delete_end (struct node * );
struct node *delete_node (struct node * );
struct node *delete_after (struct node * );
struct node *delete_left (struct node * );
struct node *sort_ll (struct node * );

int main ()
{
    int choice;
    printf ("\n++ menu ++\n1. create a list | t2\n| t3. del-beg | t4. del-end | t5. delnode | t6.\n");
}
```

\t7. del - dat \t8. sort dat \t9. exit [n])

do

3

```
print ("In enter the choice ::");
scanf ("%d", &choice);
switch (choice)
```

1

Case 1 : (start = create . 11 [start]);
printf ("In linked list created = ");
breaks;

Case 2 ; Start = display (start);

(ex 3: start = delete - beg (start))
break;

Case 4 : $\text{start} = \text{delete} - \text{end}(\text{start})$

`break;`

break;

case 6 : start = delete - after (start);
 break;

(call 2 start = delete - list (start));

break;

Case P : $t_{\text{start}} = 50 \times t - 400$ (t_{start});
break;

which (choice! = 9):

return 0;

7

struct node * delete_beg (struct node * start)

{
 struct node * ptr;
 ptr = start;
 start = start -> next;
 free (ptr);
 return start;

}

struct node * delete_end (struct node * start)

{
 struct node * ptr, * preptr;
 ptr = start;
 while (ptr -> next != NULL)
 {
 preptr = ptr;
 ptr = ptr -> next;
 }
 if (ptr -> next == NULL)
 preptr -> free (ptr);
 return start;

};

struct node * delete_node (struct node * start)

{
 struct node * ptr, * preptr;
 int val;
 printf ("In enter the value to be deleted :");
 scanf ("%d", &val);
 ptr = start;
 if (ptr -> data == val)

{

Start

Date _____
Page _____

start = delete_beg (start);
return start;

else {

 while (ptr -> next = ptr -> next);

 }

 preptr = ptr;

 ptr = ptr -> next;

 preptr -> next = ptr -> next;

 free (ptr);

 return start;

}

} ;

struct node * delete_after (struct node * start)

start struct node *ptr, *preptr;

int val;

printf ("\n enter the value to be deleted
scanf ("y. d ", &val);

ptr = start;

if (ptr -> data == val)

 start = delete_beg (start);

 return start;

 preptr = start ptr

 while (preptr -> data != val)

;);

 preptr = ptr;

 ptr = ptr -> next;

```
    pre. ptr -> next = ptr -> next;  
    free (ptr);  
    return start;  
};
```

Struct node * delete - list (Struct node * start)

```
Struct node * ptr;  
if (start != NULL);
```

```
ptr = start;  
while (ptr != NULL)
```

```
printf ("y. d is deleted \n", ptr -> data);  
start = delete (ptr);  
ptr = start;
```

```
return start;
```

```
};
```

Output:

- menu + :
- 1) create linked list
- 2) display
- 3) del - beg
- 4) del - end
- 5) del - node.
- 6) del - after
- 7) del - list

Date _____
Page _____

Enter your choice : 1

Enter -1 to exit

Enter your num : 10

Enter your num : 20

Enter your num : 30

Enter your num : 40

Enter your choice : 2

10 20 30 .

Enter your choice : 3

20

Enter your choice : 2

20 30 .

Enter your choice : 5

Enter your choice : 2

note is deleted .

✓ Your
22/11/20