ASSIGNMENT - 8

NAME : Yash Sarang

ROLL NO : 47

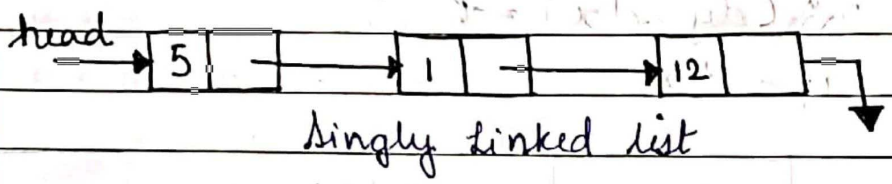
DIV : D6AD

DS EXP 6

DATE / /

AIM: To implement singly linked list ADT.

THEORY: In this type of linked list, two successive nodes of the linked list are linked with each other in sequential linear manner. Movement in forward direction is possible.



Singly linked list

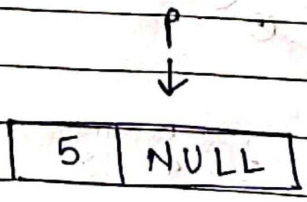
- Basic structure of a singly linked list,

```
Struct node:
{
  int data;
  struct node * next;   // Address
};
```

Memory allocation: Memory can be acquired through the use of standard library function malloc ().

Eg:

```
node *p;
p = (node*) malloc (sizeof(node));
p → data = 5;
p → next = NULL
```



Here, we declare a pointer variable P.
Size of (node) is storage requirement in no. of bytes to store a node. malloc (size of (node)) returns the address of the allocated memory block and it is assigned to variable P.
P → data = 5, stores a value of 5 in data field of node. P → next = NULL. stores a value of NULL in the next field of node.

ALGORITHM :

- For traversing a linked list :

Step 1 : [Initialize] set PTR = START
Step 2 : Repeat steps 3 and 4 while PTR != NULL
Step 3 : Apply Process to PTR DATA.
Step 4 : set PTR = PTR NEXT
(end of loop)
Step 5 : Exit.

- For counting nodes in a linked list :

Step 1 : Initialize COUNT = 0
Step 2 : If head != NULL then
temp ← head
Step 3 : while temp != NULL do
count = count + 1
temp ← temp.next
(end of loop)
Step 4 : Display count
Step 5 : Exit

PAGE No.
DATE / /

- **To search a linked list :**

Step 1 : Assign PTR = HEAD
Step 2 : Assign S = 0
Step 3 : If PTR = NULL
Step 4 : Repeat steps 5 to 7 until PTR != NULL
Step 5 : If PTR → DATA = item.
Step 6 : S = S+1
Step 7 : PTR = PTR → next
Step 8 : Exit.

Conclusion :

① At any point user can insert/delete the nodes
② When you are unpredictable about the number of elements because in linked list you no need to declare the size.
③ No need to move the elements to insert / delete.

Thus, in this way a singly linked list is implemented.

Application of linked list :

① Maintaing directory of names.
② Manipulation of polynomials by storing constants in the node of linked list.

PROGRAM CODE :

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

struct node
{
   int data;
   struct node *next;
};
struct node *start=NULL;
struct node *create_list(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 *delete_beg(struct node*);
struct node *delete_end(struct node*);
struct node *delete_node(struct node*);
struct node *delete_after(struct node*);

int main(int argc, char*argv[])
```

```
{
   int option;
   do
   {
      printf("\n\n Enter your choice :\n ");
      printf("\n 1: Create a list");
      printf("\n 2: Display a list");
      printf("\n 3: Insert a node at beginning");
      printf("\n 4: Insert a node at end");
      printf("\n 5: Insert a node bofore a given node");
      printf("\n 6: Insert a node after a given node");
      printf("\n 7: Delte a node from beginning");
      printf("\n 8: Delete a node from end");
      printf("\n 9: Delete a node before a given node");
      printf("\n 10: Delete a node after a given node");
      printf("\n 11: EXIT");

      printf("\n\nEnter your option : ");
      scanf("%d",&option);

      switch(option)
      {
      case 1:
         start=create_list(start);
         printf("\n Linked list is 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=delete_beg(start);
    break;
```

```
        case 8:
            start=delete_end(start);
            break;

        case 9:
            start=delete_node(start);
            break;

        case 10:
            start=delete_after(start);
            break;

        }
    }
    while(option !=11);
    return 0;
}

struct node *create_list(struct node *start)
{
    struct node *new_node,*ptr;
    int num;
    printf("\n Enter -1 to end");
    printf("\n Enter the data:");
```

```
    scanf("%d",&num);
    while(num!=-1)
    {
        new_node=(struct node*)malloc(sizeof(struct node));
        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("\n Enter the data:");
        scanf("%d",&num);
    }
    return start;
}

struct node *display(struct node *start)
```

```
{
   struct node *ptr;
   ptr = start;

   while(ptr !=NULL)
   {
      printf("\n %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(sizeof(struct 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, *ptr;
   int num;
   printf("\n Enter the data: ");
   scanf("%d",&num);
   new_node=(struct node*)malloc(sizeof(struct node));
   new_node->data = num;
   new_node->next=NULL;
   ptr=start;
   while(ptr->next!=NULL)
      ptr=ptr->next;
   ptr->next=new_node;

   return start;
}

struct node *insert_before(struct node *start)
{
   struct node *new_node,*ptr,*preptr;
   int num,val;
   printf("\n Enter the data: ");
   scanf("%d",&num);
   printf("\n Enter the value before which the 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;
}

struct node *insert_after(struct node *start)
{
    struct node *new_node,*ptr,*preptr;
    int num,val;
    printf("\n Enter the data: ");
    scanf("%d",&num);
    printf("\n Enter the value after which the data has to be inserted: ");
    scanf("%d",&val);
    new_node=(struct node*)malloc(sizeof(struct node));
    new_node->data = num;
    ptr=start;
```

```
    preptr=ptr;
    while(preptr->data!=val)
    {
        preptr=ptr;
        ptr=ptr->next;
    }
    preptr->next=new_node;
    new_node->next=ptr;
    return start;
}

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 *preptr, *ptr;
    ptr=start;
```

```
    while(ptr->next!=NULL)
    {
        preptr=ptr;
        ptr=ptr->next;
    }
    preptr->next=NULL;
    free(ptr);

    return start;
}

struct node *delete_node(struct node *start)
{
    struct node *preptr, *ptr;
    int val;
    printf("\n Enter the value of node which has to be deleted: ");
    scanf("%d",&val);
    ptr=start;
        if(ptr->data == val)
    {
        start= delete_beg(start);
        return start;
    }
    else
    {
```

```
        while(ptr->data!=val)
        {
           preptr=ptr;
           ptr=ptr->next;
        }
        preptr-> next = ptr -> next;
        free(ptr);
        return start;
    }
}

struct node *delete_after(struct node *start)
{
    struct node *ptr,*preptr;
    int val;
    printf("\n Enter the value after which the data has to be deleted: ");
    scanf("%d",&val);
    ptr=start;
    preptr = ptr;
    while(preptr->data!=val)
    {
        preptr=ptr;
        ptr=ptr->next;
    }
    preptr->next=ptr -> next;
```

```
    free(ptr);
    return start;
}
```

OUTPUT :

1. Creating a list

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delte a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 1

 Enter -1 to end
 Enter the data:11

 Enter the data:22

 Enter the data:33

 Enter the data:44

 Enter the data:55

 Enter the data:-1

 Linked list is created
```

2. Inserting at beginning

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 3

 Enter the data: 1


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 1
 11
 22
 33
 44
 55
```

3. Insert at end

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 4

 Enter the data: 66


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 1
 11
 22
 33
 44
 55
 66
```

4. Insert before

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 5

 Enter the data: 33

 Enter the value before which the data has to be inserted: 44


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 1
 11
 22
 33
 33
 44
 55
 66
```

5. Insert after

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 6

 Enter the data: 77

 Enter the value after which the data has to be inserted: 66


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 1
 11
 22
 33
 33
 44
 55
 66
 77
```

6. Delete at beginning

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 7


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 11
 22
 33
 33
 44
 55
 66
 77
```

7. Delete at end

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 8


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 11
 22
 33
 33
 44
 55
 66
```

8. Delete before

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 9

 Enter the value of node which has to be deleted: 44


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 11
 22
 33
 33
 55
 66
```

9. Delete after

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 10

 Enter the value after which the data has to be deleted: 33


 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 2

 11
 22
 33
 55
 66
```

10. Exit

```
 Enter your choice :

 1: Create a list
 2: Display a list
 3: Insert a node at beginning
 4: Insert a node at end
 5: Insert a node bofore a given node
 6: Insert a node after a given node
 7: Delete a node from beginning
 8: Delete a node from end
 9: Delete a node before a given node
 10: Delete a node after a given node
 11: EXIT

Enter your option : 11
```