Name :- Ajay Manohar Hirave
Roll no.:- 54 Div.:- C Batch:- C3

Experiment No. 5 :- Implementation of Double Link List and Perform Various Operations on it.

```
#include  <stdio.h>
#include <stdlib.h>
struct node
{
   struct  node  *prev;  struct  node
   *next;  int  data;  //  Added  data
   member
};
struct node *head; void
insertion_beginning(); void
insertion_last(); void
insertion_specified(); void
display(); void search(); // Added
semicolon void main()
{
   int choice = 0;
   do
   {
      printf("\n****Main Menu****\n");
      printf("\nChoose one option from the following list. ... \n");

     printf("1. insert in beginning\n2. insert at last\n3. insert at any random location\n4. search\n5.
display\n6.  Exit\n");  printf("enter
     your                choice?\n");
     scanf("\n%d",          &choice);
     switch (choice)
     {
```

```
        case 1:
        insertion_beginning();
        break; case 2:
        insertion_last(); break;
        case 3:
        insertion_specified();
        break; case 4: search();
        break; case 5: display();
        break; case 6: exit(0);
           break; default:
           printf("Please enter valid choice..");
        }
    } while (choice != 6);

}

void insertion_beginning()
{
    struct node *ptr; int item; ptr = (struct node
    *)malloc(sizeof(struct node)); if (ptr == NULL)
    {
        printf("\nOVERFLOW");

    }
    else
    {
        printf("\nEnter item value");
        scanf("%d", &item);

        if (head == NULL)
        {
```

```
            ptr->next = NULL;
            ptr->prev = NULL;
            ptr->data = item;
            head = ptr;
        }
        else
        {
            ptr->data = item;
            ptr->prev = NULL;
            ptr->next = head;
            head->prev = ptr;
            head = ptr;
        }
        printf("\nNode inserted\n");
    }

}

void insertion_last()
{
    struct node *ptr, *temp; int item; ptr = (struct
    node *)malloc(sizeof(struct node)); if (ptr ==
    NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter item value"); scanf("%d",
        &item);
        ptr->data = item;
```

```
        if (head == NULL)
        {
           ptr->next = NULL; ptr-
           >prev = NULL; head =
           ptr;
        }
        else
        {
           temp = head; while (temp-
           >next != NULL)
           {
              temp = temp->next;
           }
           temp->next = ptr;
           ptr->prev = temp;
           ptr->next = NULL;
        }
     }
     printf("\nNode inserted\n");

}
void insertion_specified()
{
     struct node *ptr, *temp;
     int item, loc, i; ptr = (struct node
     *)malloc(sizeof(struct node)); if (ptr == NULL)
     {
        printf("\nOVERFLOW");
     }
     else
```

```
    {
       temp = head; printf("\nEnter
       the location"); scanf("%d",
       &loc);
       for (i = 0; i < loc; i++)
       {
          temp = temp->next; if
          (temp == NULL)
          {
             printf("\n There are less than %d elements", loc); return;
          }
       }
       printf("enter value");
       scanf("%d", &item); ptr-
       >data = item; ptr->next =
       temp->next; ptr->prev =
       temp; temp->next = ptr;
       temp->next->prev = ptr;
       printf("\nnode inserted\n");
       }

}

void display()
{
    struct node *ptr; printf("\n
    printing values..\n"); ptr =
    head; while (ptr != NULL)
    {
       printf("%d\n", ptr->data); ptr
       = ptr->next;
```

```
    }
}

void search()
{
    struct node *ptr; int
    item, i = 0, flag;
    ptr  =  head;  if
    (ptr == NULL)
    {
        printf("\nempty list\n");
    }
    else
    {
        printf("\nEnter item which you want to search?\n");
        scanf("%d", &item); while (ptr != NULL)
        {
            if (ptr->data == item)
            {

                printf("\nitem found at location %d", i + 1);
                flag = 0; break;


            }
            else
            { flag = 1;
            }
            i++;  ptr  =  ptr-
            >next;


        }
```

```
        if (flag == 1)
        {
           printf("\nItem not found\n");
        }
}

 }
```

OUTPUT :-

****Main Menu****

Choose one option from the following list....

1. insert in beginning

2. insert at last

3. insert at any random location

4. search

5. display 6. Exit enter your choice?

1

Enter item value 10

Node inserted

****Main Menu****

Choose one option from the following list....

1. insert in beginning

2. insert at last

3. insert at any random location

4. search

5. display 6. Exit enter your choice? 2

Enter item value 20

Node inserted

****Main Menu****

Choose one option from the following list....

1. insert in beginning

2. insert at last

3. insert at any random location

4. search

5. display 6. Exit enter your choice?

3

Enter the location 30

There are less than 30 elements

****Main Menu****

Choose one option from the following list....

1. insert in beginning

2. insert at last

3. insert at any random location

4. search

5. display 6. Exit enter your choice?

4

Enter item which you want to search?

10 item found at

location 1