PROGRAM 5

Write a program to Implement Singly Linked List with following operations:a) Create a linked list. b) Insertion of a node at first position, at any position and at end of list. c) Display the contents of the linked list.

```c
#include <stdlib.h>
#include <stdio.h>
void create();
void display();
void insert_begin();
void insert_end();
void insert_pos();
struct node
{
    int info;
    struct node *next;
};
struct node *start = NULL;
int main()
{
    int choice;
    while (1)
    {

        printf("\n MENU           \n");
        printf("\n 1.Create     \n");
        printf("\n 2.Display    \n");
```

```c
        printf("\n 3.Insert at the beginning    \n");
        printf("\n 4.Insert at the end  \n");
        printf("\n 5.Insert at specified position
\n");
        printf("\n 6.Exit         \n");
        printf("\n------------------------------------
\n");
        printf("Enter your choice:");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            create();
            break;
        case 2:
            display();
            break;
        case 3:
            insert_begin();
            break;
        case 4:
            insert_end();
            break;
        case 5:
            insert_pos();
            break;
        case 6:
            exit(0);
            break;
```

```c
        default:
            printf("\n Wrong Choice:\n");
            break;
        }
    }
    return 0;
}
void create()
{
    struct node *temp, *ptr;
    temp = (struct node *)malloc(sizeof(struct node));
    if (temp == NULL)
    {
        printf("\nOut of Memory Space:\n");
        exit(0);
    }
    printf("\nEnter the data value for the node:");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if (start == NULL)
    {
        start = temp;
    }
    else
    {
        ptr = start;
        while (ptr->next != NULL)
        {
            ptr = ptr->next;
```

```c
        }
        ptr->next = temp;
    }
}
void display()
{
    struct node *ptr;
    if (start == NULL)
    {
        printf("\nList is empty:\n");
        return;
    }
    else
    {
        ptr = start;
        printf("\nThe List elements are:\n");
        while (ptr != NULL)
        {
            printf("%d", ptr->info);
            ptr = ptr->next;
        }
    }
}
void insert_begin()
{
    struct node *temp;
    temp = (struct node *)malloc(sizeof(struct node));
    if (temp == NULL)
    {
```

```c
        printf("\nOut of Memory Space:\n");
        return;
    }
    printf("\nEnter the data value for the node:");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if (start == NULL)
    {
        start = temp;
    }
    else
    {
        temp->next = start;
        start = temp;
    }
}
void insert_end()
{
    struct node *temp, *ptr;
    temp = (struct node *)malloc(sizeof(struct node));
    if (temp == NULL)
    {
        printf("\nOut of Memory Space:\n");
        return;
    }
    printf("\nEnter the data value for the node:");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if (start == NULL)
```

```c
    {
        start = temp;
    }
    else
    {
        ptr = start;
        while (ptr->next != NULL)
        {
            ptr = ptr->next;
        }
        ptr->next = temp;
    }
}
void insert_pos()
{
    struct node *ptr, *temp;
    int i, pos;
    temp = (struct node *)malloc(sizeof(struct node));
    if (temp == NULL)
    {
        printf("\nOut of Memory Space:\n");
        return;
    }
    printf("\nEnter the position for the new node to be inserted:");
    scanf("%d", &pos);
    printf("\nEnter the data value of the node:");
    scanf("%d", &temp->info);
```

```c
        temp->next = NULL;
    if (pos == 0)
    {
        temp->next = start;
        start = temp;
    }
    else
    {
        for (i = 0, ptr = start; i < pos - 1; i++)
        {
            ptr = ptr->next;
            if (ptr == NULL)
            {
                printf("\nPosition not found:\n");
                return;
            }
        }
        temp->next = ptr->next;
        ptr->next = temp;
    }
}
```

Output –

```
                MENU
 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:1

Enter the data value for the node:22
                MENU
 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:4

Enter the data value for the node:44
```

```
                MENU
 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:3

Enter the data value for the node:11
                MENU
 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:5

Enter the position for the new node to be inserted:2
```

```
 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:2

The List elements are:
11223344
                MENU

 1.Create

 2.Display

 3.Insert at the beginning

 4.Insert at the end

 5.Insert at specified position

 6.Exit
--------------------------------------
Enter your choice:6


...Program finished with exit code 0
Press ENTER to exit console.
```