**DSA PROGRAMMING LAB-4**

**Link List**

• Name :HITU RAJ

• Roll no. :2005025

• Branch :CSE

/\* Q1.WAP to create SLLand display the value of each node

 using user defined funcion create and display\*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data\_025;

    struct node \*next\_025;

} \* first\_025, \*temp\_025, \*last\_025;

void create(int a[10], int n)

{

    first\_025 = (struct node \*)malloc(sizeof(struct node));

    first\_025->data\_025 = a[0];

    first\_025->next\_025 = NULL;

    last\_025 = first\_025;

    for (int i = 1; i < n; i++)

    {

        temp\_025 = (struct node \*)malloc(sizeof(struct node));

        temp\_025->data\_025 = a[i];

        temp\_025->next\_025 = NULL;

        last\_025->next\_025 = temp\_025;

        last\_025=temp\_025;

    }

}

void display(struct node \*F)

{

    while (F!=NULL)

    {

        printf("%d ", F->data\_025);

        F = F->next\_025;

    }

}

int main()

{

    int a[] = {1, 2, 3, 4, 5, 6, 7};

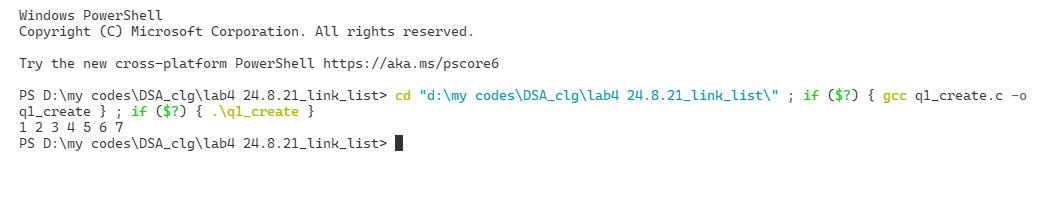
    create(a, 7);

    display(first\_025);

    return 0;

}

OUTPUT-Q1



/\* Q2.WAP to create SLL using recursion and display the value of each node

 using user defined funcion create and display\*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data\_025;

    struct node \*next\_025;

} \* first\_025, \*temp\_025, \*last\_025;

void crearec(int a[], int n, int i)

{

    if (i == n)

    {

        printf("nodes are created\n");

    }

    else

    {

        struct node \*temp\_025;

        temp\_025 = (struct node \*)malloc(sizeof(struct node));

        temp\_025->data\_025 = a[i];

        temp\_025->next\_025 = NULL;

        last\_025->next\_025 = temp\_025;

        last\_025 = temp\_025;

        crearec(a, n, i + 1);

    }

}

void display(struct node \*f)

{

    //struct node \*temp\_025 = f;

    while (f != NULL)

    {

        printf("%d ", f->data\_025);

        f = f->next\_025;

    }

}

int main()

{

    int a[] = {1, 2, 3, 4, 5, 6, 7};

    first\_025 = (struct node \*)malloc(sizeof(struct node));

    first\_025->data\_025 = a[0];

    first\_025->next\_025 = NULL;

    last\_025 = first\_025;

    crearec(a, 7, 1);

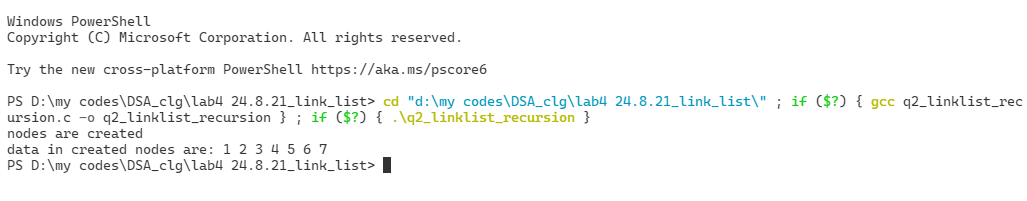
    printf("data in created nodes are: ");

    display(first\_025);

    return 0;

}

OUTPUT-Q2



/\*Q3.inserting new node at beginning

inserting new node at end

inserting new node at any place\*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data\_025;

    struct node \*next\_025;

} \* first\_025, \*last\_025, \*temp\_025;

void create(int a[10], int n)

{

    first\_025 = (struct node \*)malloc(sizeof(struct node));

    first\_025->data\_025 = a[0];

    first\_025->next\_025 = NULL;

    last\_025 = first\_025;

    for (int i = 1; i < n; i++)

    {

        temp\_025 = (struct node \*)malloc(sizeof(struct node));

        temp\_025->data\_025 = a[i];

        temp\_025->next\_025 = NULL;

        last\_025->next\_025 = temp\_025;

        last\_025 = temp\_025;

    }

}

void display(struct node \*temp\_025)

{

    while (temp\_025 != NULL)

    {

        printf("%d ", temp\_025->data\_025);

        temp\_025 = temp\_025->next\_025;

    }

}

void insertbeg(struct node \*temp\_025)

{

    struct node \*newnode;

    printf("entre the data to the node ");

    newnode = (struct node \*)malloc(sizeof(struct node));

    scanf("%d", &newnode->data\_025);

    newnode->next\_025 = temp\_025;

    first\_025 = newnode;

}

void insertlast(struct node \*temp\_025)

{

    int a;

    printf("entre the data to the node");

    scanf("%d", &a);

    struct node \*newnode;

    newnode = (struct node \*)malloc(sizeof(struct node));

    newnode->data\_025 = a;

    newnode->next\_025 = NULL;

    temp\_025->next\_025 = newnode;

    last\_025 = newnode;

}

void insertany(struct node \*temp\_025, int n)

{

    int a;

    printf("entre the data to the node");

    scanf("%d", &a);

    struct node \*newnode;

    newnode = (struct node \*)malloc(sizeof(struct node));

    newnode->data\_025 = a;

    newnode->next\_025 = NULL;

    for (int i = 1; i < n; i++)

    {

        temp\_025 = temp\_025->next\_025;

    }

    newnode->next\_025 = temp\_025->next\_025;

    temp\_025->next\_025 = newnode;

}

int main()

{

    int n,p;

    int a[] = {1, 2, 3, 4, 5, 6, 7};

    create(a, 7);

    display(first\_025);

    printf("\n press 1 to insert in begin\n press 2 to insert in last \npress 3 to insert b/w any node\n");

    scanf("%d", &n);

    switch (n)

    {

    case 1:

        insertbeg(first\_025);

        display(first\_025);

        break;

    case 2:

        insertlast(last\_025);

        display(first\_025);

        break;

    case 3:

        printf("After which node you want to enter");

        scanf("%d", &p);

        insertany(first\_025,p);

        display(first\_025);

        break;

    default:

        break;

    }

    return 0;

}

OUTPUT-Q3

