Program 8

Write a Program to Implement doubly link list with primitive operations

- a) Create a doubly linked list.
- b) Insert a new node at the beginning.
- c) Insert the node based on a specific location
- d) Insert a new node at the end.
- e) Display the contents of the list

Observation:

```
WAP to emplement doubly lented lest and
i) insertion at the beginning.
ii) insertion at the end
   PER) inserteon at the specific position
# Enclude (stdlo.h)
# Enclude (stalleb.h)
statet node f (x this) post poor box
   strut node" prev
    strut node next;
struct node *head, *tail;
void create Die ()
 E staut node* newnode;
    head = tail = NULL;
    int choice 21;
     while (choice) - Jan + Jan + January
     new node 2 (struct node *) malloc (size of (struct now)
     pelat f ("Enter the data").
     scanf (".1.d", newnode -> data);
      newnode -> next = NULL;
      newnode -> prev = NULL:
      if (head = = NULL)
      head = tail = new node;
        tall-) next = new node;
        new node-) prev = tal;
```

```
tael z new node;
 }
prentf ("Enter 1" to continue:");
    scanf (" 1.d" & chorce);
Void EnsertBeg (int x)
    struit node" new node;
     newnoole 2 (steert noole*) malloc (stree) f(steert
     newnode -) data = x; 1 + x = 1 + x = 1
     new node - previ scott;
      if (head = NULL) into
         head 2 tail 2 nuvnode; 1919
      clse
       Enewnode + next = head;
       new head-) prev z new node;
   stant node * newhode;
   newnode 2 (struct mode ») mallo c (streof (struct
    new node -> data = x:
    new node - next. NULL;
```

```
of (head = 2 NULL)
    hende tall : newnode; warmen
    newnod - prever NULLY - VARGE- JANK - Drowners
 else
  { nownode + prev = tall;
    tail - next = new node;
     tael = newnode
void enserthou (entx)
   steuct node" nownode, "temp;
    Ent pos-
    prenty ("Enter the posetton");
     scanf (" .t.d", f pos);
     if (pos=21)
       void Privot Beg(x);
     else
     newnode = (structnode*) malloc (streof (structnode));
     newnode -> data = x;
      newnode -> prev= NULC;
      newnode - next: NULL;
      temps head;
      for ( 12 te 1, 2 < pos -1; 1++)
       E temp = temp = next;
         it (temp==tall) next)
          I paint ("There are less than und modes", Dos);
          3 return;
```

```
new node -> priv = temp -> next;

rewnoode -> next = new node;

new node -> next -> prev = new node;

new node -> next -> prev = new node;

Per new node -> next -> prev = new node;

Set to the data: 1

Enter the data: 2

enter 1 to continue: 1

Enter the data: 3

enter 1 to continue: 1

Enter the data: 3

enter 1 to continue: 1

Insert at Eggenning

a Insert at Specific position

Insert at End

U Display List

Enter value to Priest at the beginning at

Menu:

1. Insert at Segenning

a Frient at Segenning
```

```
Menu:
1. Insert at Beginning
2. Insert at specific position
   3. Insert at End
  u. Display lext
     Enter your choices 
Enter value to Prosert: 5
Enter the postfons 2
  Monu:

J. Insert at Beginning

a. Insert at specific posetton

E. Insert at Ford

Details of the series of the ser
   4. Display list
5. Exet
    s. Exit
Enter your choice: 4
4+5+1+2+3+>NULL
1. Ensert at Beginning
a. Insert at specific Position

3. Insert at End

The Position Pest
4. In Display lest
s. Exit
  5. Exit
Enter your choice: 3
Enter value to insert at the end: 6
   Menu:
 1. Insert at Beginning
a. Insert at Specific Position)
3. Insert at End
   4. Display lest
  5. Exit.
   Enter your choice: 4.
    4 (>5 (>) 1 (> 2 (-> 3 (-> 6 (-> NULL
```

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct node {
   int data;
    struct node *next;
    struct node *prev;
};
struct node *head, *tail;
void createDLL(){
    struct node *newnode;
    head = tail = NULL;
    int choice = 1;
    while(choice==1){
        newnode = (struct node *)malloc(sizeof(struct node));
        printf("Enter the data:");
        scanf("%d", &newnode->data);
        newnode->next = NULL;
        newnode->prev = NULL;
        if(head == NULL){
        head = tail = newnode;
        else{
        tail->next = newnode;
        newnode->prev = tail;
        tail = newnode;
        printf("enter 1 to continue:");
        scanf("%d", &choice);
void insertBeg(int x){
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->prev = NULL;
    if(head == NULL){
    head = tail = newnode;
    newnode->next = NULL;
    else{
    newnode->next = head;
    head->prev = newnode;
    head = newnode;
void insertEnd(int x){
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
```

```
newnode->data = x;
    newnode->next = NULL;
    if(head == NULL){
    head = tail = newnode;
    newnode->prev = NULL;
    else{
    newnode->prev = tail;
    tail->next = newnode;
    tail = newnode;
void insertPos(int x){
    struct node *newnode, *temp;
    int pos;
    printf( "Enter the position:");
    scanf("%d", &pos);
    if(pos == 1){
        insertBeg(x);
    else{
        newnode = (struct node *)malloc(sizeof(struct node));
        newnode->data = x;
        newnode->next = NULL;
        newnode->prev = NULL;
        temp = head;
        for(int i = 1; i < pos-1; i++){
            temp=temp->next;
            if(temp == tail->next){
                printf("There are less than %d nodes\n",pos);
                return;
        }
        newnode->prev = temp;
        newnode->next = temp->next;
        temp->next = newnode;
        newnode->next->prev = newnode;
void display(){
    struct node *temp;
    if(head == NULL){
        printf("list is empty\n");
    else{
        temp = head;
        while(temp != NULL){
            printf("%d<->", temp->data);
            temp = temp->next;
```

```
printf("NULL\n");
int main() {
createDLL();
 int choice, value, position;
    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert at Beginning\n");
        printf("2. Insert at Specific Position\n");
        printf("3. Insert at End\n");
        printf("4. Display List\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter value to insert at the beginning: ");
                scanf("%d", &value);
                insertBeg(value);
                break;
            case 2:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                insertPos(value);
                break;
            case 3:
                printf("Enter value to insert at the end: ");
                scanf("%d", &value);
                insertEnd(value);
                break;
            case 4:
                display();
                break;
            case 5:
                printf("Exiting...\n");
                return 0;
            default:
                printf("Invalid choice, please try again.\n");
    return 0;
```

Output:

Menu:

- 1. Insert at Beginning
- 2. Insert at Specific Position
- 3. Insert at End
- 4. Display List
- 5. Exit

Enter your choice: 3

Enter value to insert at the end: 6

Menu:

- Insert at Beginning
- 2. Insert at Specific Position
- 3. Insert at End
- 4. Display List
- Exit

Enter your choice: 4

4<->5<->1<->2<->3<->6<->NULL