Program 7

Write A Program to Implement doubly link list with primitive operations

- a) Create a doubly linked list.
- b) Insert a new node to the left of the node.
- c) Delete the node based on a specific value
- d) Display the contents of the list

```
Code:
```

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* prev;
  struct Node* next;
};
struct Node* createNode(int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = data;
  newNode->prev = NULL;
  newNode->next = NULL;
  return newNode;
}
void insertAtBeginning(struct Node** head, int data) {
  struct Node* newNode = createNode(data);
  if (*head == NULL) {
     *head = newNode;
  } else {
     newNode->next = *head;
     (*head)->prev = newNode;
     *head = newNode;
}
```

```
void insertAtPosition(struct Node** head, int data, int position) {
  if (position < 1) {
    printf("Invalid position!\n");
    return;
  }
  struct Node* newNode = createNode(data);
  if (position == 1) {
    insertAtBeginning(head, data);
    return;
  }
  struct Node* temp = *head;
  for (int i = 1; temp != NULL && i < position - 1; i++) {
    temp = temp->next;
  }
  if (temp == NULL) {
    printf("Position out of bounds!\n");
    free(newNode);
    return;
  }
  newNode->next = temp->next;
  newNode->prev = temp;
  if (temp->next != NULL) {
    temp->next->prev = newNode;
  temp->next = newNode;
}
void insertAtEnd(struct Node** head, int data) {
  struct Node* newNode = createNode(data);
  if (*head == NULL) {
    *head = newNode;
    return;
  }
  struct Node* temp = *head;
```

```
while (temp->next != NULL) {
     temp = temp->next;
  temp->next = newNode;
  newNode->prev = temp;
}
void displayList(struct Node* head) {
  if (head == NULL) {
     printf("List is empty!\n");
     return;
  }
  struct Node* temp = head;
  printf("List contents: ");
  while (temp != NULL) {
     printf("%d", temp->data);
     temp = temp->next;
  }
  printf("\n");
}
int main() {
  struct Node* head = NULL;
  int choice, data, position;
  while (1) {
     printf("\nDoubly Linked List Operations:\n");
     printf("1. Insert at Beginning\n");
     printf("2. Insert at 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 data to insert at beginning: ");
          scanf("%d", &data);
```

```
insertAtBeginning(&head, data);
       break;
     case 2:
       printf("Enter data to insert: ");
       scanf("%d", &data);
       printf("Enter position: ");
       scanf("%d", &position);
       insertAtPosition(&head, data, position);
       break;
    case 3:
       printf("Enter data to insert at end: ");
       scanf("%d", &data);
       insertAtEnd(&head, data);
       break;
     case 4:
       displayList(head);
       break;
     case 5:
       printf("Exiting program.\n");
       exit(0);
     default:
       printf("Invalid choice! Please try again.\n");
  }
return 0;
```

Doubly Linked List Operations: 1. Insert at Beginning 2. Insert at Position 3. Insert at End 4. Display List Exit Enter your choice: 1 Enter data to insert at beginning: 1 Doubly Linked List Operations: 1. Insert at Beginning 2. Insert at Position 3. Insert at End 4. Display List Exit Enter your choice: 2 Enter data to insert: 2 Enter position: 2 Doubly Linked List Operations: 1. Insert at Beginning 2. Insert at Position Insert at End 4. Display List Exit Enter your choice: 3 Enter data to insert at end: 3

Doubly Linked List Operations:

1. Insert at Beginning

2. Insert at Position

3. Insert at End

4. Display List

5. Exit
Enter your choice: 4
List contents: 1 2 3

Doubly Linked List Operations:

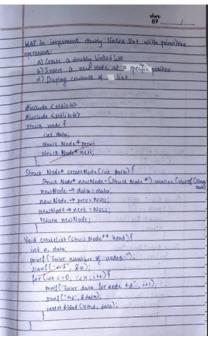
1. Insert at Beginning

2. Insert at Position

3. Insert at End

4. Display List

5. Exit
Enter your choice: 5



Nord harnetter (since there * had, he does) 1
Since Nore * weather scenes Nord Chains,
if China = NOVE)
* had > non North; else { new Node + next - + head; (" head) - prer = new Nodes * head = new Nede; void insert At Pas (struct Node >+ head, int data, int pas) orine ("Invalid position ! In); The relient Struck Node+ newblode-cremeNode (said); if (posses = 1) { issen & Bry (newly data); . . . Ytlam; Struck Node + Jours Mundi for City is by Jemph = NULL & B i & position - 19: 114) } lemp : Jemp -> next; if Clour - Nuic) } grint ("Position out of bounds: ""); Lyce (new Node); Ttlam;

Exiting program.

50m.	and Pro
newbless - neet = leme - neet;	World maint X
as a de anni a level;	struck Node* handensule
if Champa next! = Now!) ?	int chelie idala, cos,
Jemp-nere - prevo hew Node;	The state of the s
A CONTRACTOR OF THE PARTY OF TH	pirth Denky lines list Operations: \");
world lustre HEND (Struck Node ** mone wint dala) 5	print to crow with last last last last last last last last
Struck Noole* new Noole's create Noole (dosa);	print 11"2 hears at Beginning In"
il (head = NULL) &	County of District of District of Or
"hens " New Node :	print("5. Insur at position 14");
YChow;	profit 4 Insur at end In")
	print(=5. Dieplay list 14");
struct Noder Jemp - Motord;	print ("b. Exit \un");
while Clemp - Antre 1: Novel)	print(" East your clubice:");
temp > temp = ned;	grant ("4.5", & cherre)
TOOK TOWN THE TO	awitch Cohoice of the
Jemp Truck snewHode'.	Case 1 : Corate Lies (Alward)
	Lev many many many many
new Node - priva long;	case 2: print ("Euler data to invest at beginning:")
24	man & (" of a", sedata);
void display (strice Node + hera) }	Good to Bry Chandendali
if Cheat == NULL)	Oraki
print ("Emply list");	tox 8: Diver ("Luker data to ivanet");
Yelsow.	Scanif "Hod", forda);
	grint ("Introosition:");
Struck Nodes semps hept;	goul ("Ad", Ares)
MINI ("L'S CONTENS:");	inscript Pos Calvent, data, Desilver 1
white Champ!= NULL)!	brox:
print ("des", sung-posterio	case to: privat ("fuzzy data to insert at end in)
Jemp: Jemp-s neat;	grant ("del", helala);
Carlotte Market Connection	inter to End (Phea), data).
perel ("\n");	breat.
A Carll mark a server	case 5; disday (lead);
The state of the s	brook)
	OX(0K)

nds .	
10	~
	-
upid main()	
Struck Node* hoods Nive	
int thelice idata, pos.	
white(1)?	
pint ("Deaky Linked list Operations: \");	
print [1"2 beent at Beginning In	
print(1" 5. Insur at position in)	
profit 4. Insert at sud In")	
print (" b. Exit \ \n");	
print(" b. Exit war choice.");	
grant ("46", & cherce);	
granten Choice H	
Car I : (mose list (Shead))	
See V. Control of the Party of	
case 2: print ("There data to limest at beginning:	1
Sant (42, Fante)	-
insert of the Cheen to day a ?	
break)	-
tax 3: prive ("Luki dala le ivant");	-
Scan [[" Hod", benden);	-
print("Inter position:")	-
soul ("-13", knos)	-
juscript Pos Caland, dala, position is	7
cost is private "factor data to inject of end int	1
grant ("del", hola);	-
inter to Good (Thea), data)	П
break.	
case 5; disday (lead);	
	-

	store 67
-	
-	Can 6: printf ("Exiting pageaus in");
_	evit(e);
_	can 6: prints ("Invalid choice !"); defaue: iprints ("Invalid choice !");
_	OCT TO THE PARTY OF THE PARTY O
	3
	Output:
	Pourly Linked Liet Operations:
	1. Insert or Beginning
	2. Insurt at Position
_	3. I want at Eud
	4 · Distantist
	4. Display list 5. Exil
	Duler year Chaice : 1
	Europala to insert at Buginsming:
	Enter Yours distre: 2
	Fully data to fourt : 2
	Culty position 2
	Ener your chaire 3
	Gets dola to justif all end: 5
	fuer year choice: tr
	List Pourus: 1 2 2
	" W.
Va	No Maria
1	