Program 4

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.

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
  struct Node* next;
};
void createList(struct Node** head);
void insertAtBeginning(struct Node** head, int data);
void insertAtPosition(struct Node** head, int data, int position);
void insertAtEnd(struct Node** head, int data);
void displayList(struct Node* head);
int main() {
  struct Node* head = NULL;
  int choice, data, position;
  while (1) {
     printf("\nMenu:\n");
     printf("1. Create a linked list\n");
     printf("2. Insert at the beginning\n");
     printf("3. Insert at a specific position\n");
     printf("4. Insert at the end\n");
     printf("5. Display the list\n");
     printf("6. Exit\n");
     printf("Enter your choice: ");
     scanf("%d", &choice);
     switch (choice) {
       case 1:
          createList(&head);
          break:
       case 2:
          printf("Enter data to insert at the beginning: ");
          scanf("%d", &data);
          insertAtBeginning(&head, data);
          break;
       case 3:
          printf("Enter data to insert: ");
          scanf("%d", &data);
```

```
printf("Enter position to insert (starting from 1): ");
          scanf("%d", &position);
          insertAtPosition(&head, data, position);
          break:
       case 4:
          printf("Enter data to insert at the end: ");
          scanf("%d", &data);
          insertAtEnd(&head, data);
          break;
       case 5:
          displayList(head);
          break;
       case 6:
          printf("Exiting the program.\n");
          exit(0);
       default:
          printf("Invalid choice. Please try again.\n");
     }
  }
  return 0;
}
void createList(struct Node** head) {
  int data, choice;
  do {
     printf("Enter data to insert: ");
     scanf("%d", &data);
     insertAtEnd(head, data);
     printf("Do you want to add another node? (1 for Yes, 0 for No): ");
     scanf("%d", &choice);
  } while (choice != 0);
}
void insertAtBeginning(struct Node** head, int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
     printf("Memory allocation failed.\n");
     return;
  newNode->data = data;
  newNode->next = *head;
  *head = newNode;
  printf("Node inserted at the beginning.\n");
void insertAtPosition(struct Node** head, int data, int position) {
  if (position < 1) {
     printf("Invalid position.\n");
     return;
```

```
}
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
    printf("Memory allocation failed.\n");
  newNode->data = data;
  if (position == 1) {
    newNode->next = *head;
    *head = newNode;
    printf("Node inserted at position %d.\n", position);
    return;
  }
  struct Node* temp = *head;
  for (int i = 1; i < position - 1 && temp != NULL; <math>i++) {
    temp = temp->next;
  if (temp == NULL) {
    printf("Position out of bounds.\n");
    free(newNode);
    return;
  }
  newNode->next = temp->next;
  temp->next = newNode;
  printf("Node inserted at position %d.\n", position);
}
void insertAtEnd(struct Node** head, int data) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  if (newNode == NULL) {
    printf("Memory allocation failed.\n");
    return;
  newNode->data = data;
  newNode->next = NULL;
  if (*head == NULL) {
    *head = newNode;
  } else {
    struct Node* temp = *head;
    while (temp->next != NULL) {
       temp = temp->next;
    temp->next = newNode;
  }
```

```
printf("Node inserted at the end.\n");
}

void displayList(struct Node* head) {
    if (head == NULL) {
        printf("The list is empty.\n");
        return;
    }

printf("Linked list contents: ");
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

```
Menu:

1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 2
Enter data to insert at the beginning: 12
Node inserted at the beginning.

Menu:
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 2
Enter data to insert at the beginning: 23
Node inserted at the beginning.

Menu:
1. Create a linked list
2. Insert at the beginning.

Menu:
1. Create a linked list
2. Insert at a specific position
4. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 3
Enter data to insert: 2
Enter position to insert (starting from 1): 2
Node inserted at position 2.
```

```
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 5
Linked list contents: 23 -> 2 -> 12 -> NULL
1. Create a linked list
2. Insert at the beginning
3. Insert at a specific position
4. Insert at the end
5. Display the list
6. Exit
Enter your choice: 6
Exiting the program.
```

| 05 a) WAP to Implement Singly Linked Lut ux following operations | |
|--|--------------------------------------|
| | *herd = resiliate; |
| - coopts a linked like | 3 toh a- quat " a- 1 X 1 I Reman |
| investion of a node at tout position and | Sound Node * Jemp = * head; |
| at and a the list | 187 (M) L= 1: [2 MX-7 2 to 1 |
| Display The contents of linked lits | 3 stemp = stemp = next; |
| | |
| # include < stdio.h> | 3 (JUN == great) fi |
| #include < stdlib.h > | & bonuf (, Bornion on a souds /v.) |
| * + 11-C | the first of the sale (13 sale 6). |
| stourt Node & | John Whode - Clark |
| int data; Sourt Note * next; | reutlede - next = temp - rext |
| 2 Sound Name * TRAL; | 3 temp -> next = newNode; |
| 0; | 3 |
| Struct Node * co-eateNode (int data) { | bead ** shall brist a treating bigs |
| struct Node * penallate = (structMarte *) mallar | So toto toto |
| (stall trunt) page) | Metall = shall all * shall trundle |
| newNode -> data = data; | (data) |
| newNode → rext = NULL; | 3 (* head == NULL) & |
| statusm neurNorte; | * head = newNede; |
| 3 | . newser. |
| WHAT IS NOT THE REAL PROPERTY. | 3 turning |
| word invest at beginning (Jewat Node ** head, sit of) | board * = gomet * about truets |
| struct Node * new Node = OverleNode (data); | ishale (temp -> next! = Now)? |
| rountede -> next = * head; | ; tren - groupt = great |
| * head = newNode; | 3 tul gat makers |
| | temp → next = new Morter; |
| | 3 to gold man patrice |
| By the the teach the state of the second bis | Ched * shall trude I work they read |
| The mental product source the sale in the sale is | 4 4 (head == MULL) & |
| 3 (pos == 1) & intpos) & | week to the print (2 mptg (1 th)) |

| Date | |
|---|--|
| Just Nade * temp = head; while (temp) = Null) ? pointly ('/ d -> , temp - data). temp = temp -> rest; | Dots Page |
| Source Node * temp = (Veres) | Sinted Just chaice: 3 Jinhad Just 12 - 23 - NULL |
| while (temp! = NOLL) | Jinhod Just 1112 - 23 - NULL |
| point town - rext | Cate was chart t |
| Jemp = itship | enter your choice: 1 |
| 3 point ("NUL In"); | |
| 3 10000 | Enter your chaice: 2 |
| 3 man () Emport) 11 | 2011es : 568 |
| July Willy * head = NULL | and the same of th |
| | Enter your choice 3 |
| invest at beginning (spead, 5); display (head); | enter: If the empty of |
| display (head); | Jinked Jist: 98 -> 12 -> 23 -> 568 |
| - MARANT - 100 1 15 2) 8 | THE STREET PORT SHAPE - NULL |
| invest at post (fread, 15,3); duplay (read); | S(1-12 = 1 Prost land 14 " C" = - [1] =) |
| diplay (head); | O & Con |
| short at end (shend, 30); | O P |
| duplay (head) with truth | |
| auplay (read) | |
| endium O; | |
| 3 | My as |
| mentale 8 | 2/3 |
| | 13/24 |
| ait to a good of stold touch | |
| . That at the beginning | And the second of the second o |
| Trest at the beginning a. Trest at the end | |
| a. Insed at the site | |
| 3. Trylay the Jut | |
| 4. 200 | |
| Enter your choice: | |
| Enter value to invest at the beginning to | THE RESERVE THE PROPERTY OF THE PARTY OF THE |
| SOJUM == hoard) (c) | and the second second second second |
| Entor your choice: 2 mg | |
| Enter the value to insert at the end 28 | |
| | The second secon |