/* WAP 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. Display the contents of the linked list. */

```
#include <stdio.h>
#include <stdlib.h>

struct node {
   int info;
   struct node* next;
};

struct node* createlk() {
   struct node* p;
   struct node* start = NULL;
   struct node* last = NULL;
   int item;

   printf("Enter elements (-999 to stop):\n");
   scanf("%d", &item);

   while (item != -999) {
      p = (struct node*)malloc(sizeof(struct node));
      p->info = item;
      p->next = NULL;

      if (start == NULL) {
```

```
            start = p;
            last = p;
        } else {
            last->next = p;
            last = p;
        }
        scanf("%d", &item);
    }
    return start;
}

struct node* inserfirst(struct node* start, int item) {
    struct node* p = (struct node*)malloc(sizeof(struct node));
    p->info = item;
    p->next = start;
    return p;
}

struct node* insertlast(struct node* start, int item) {
    struct node* p = (struct node*)malloc(sizeof(struct node));
    struct node* last;

    p->info = item;
    p->next = NULL;
```

```
    if (start == NULL) {
        return p;
    }

    last = start;
    while (last->next != NULL) {
        last = last->next;
    }

    last->next = p;
    return start;
}

struct node* insertatpositon(struct node* start, int item, int pos) {
    struct node *p, *temp;
    p = (struct node*)malloc(sizeof(struct node));
    p->info = item;

    if (pos == 1) {
        p->next = start;
        return p;
    }

    temp = start;
    int size = 1;
```

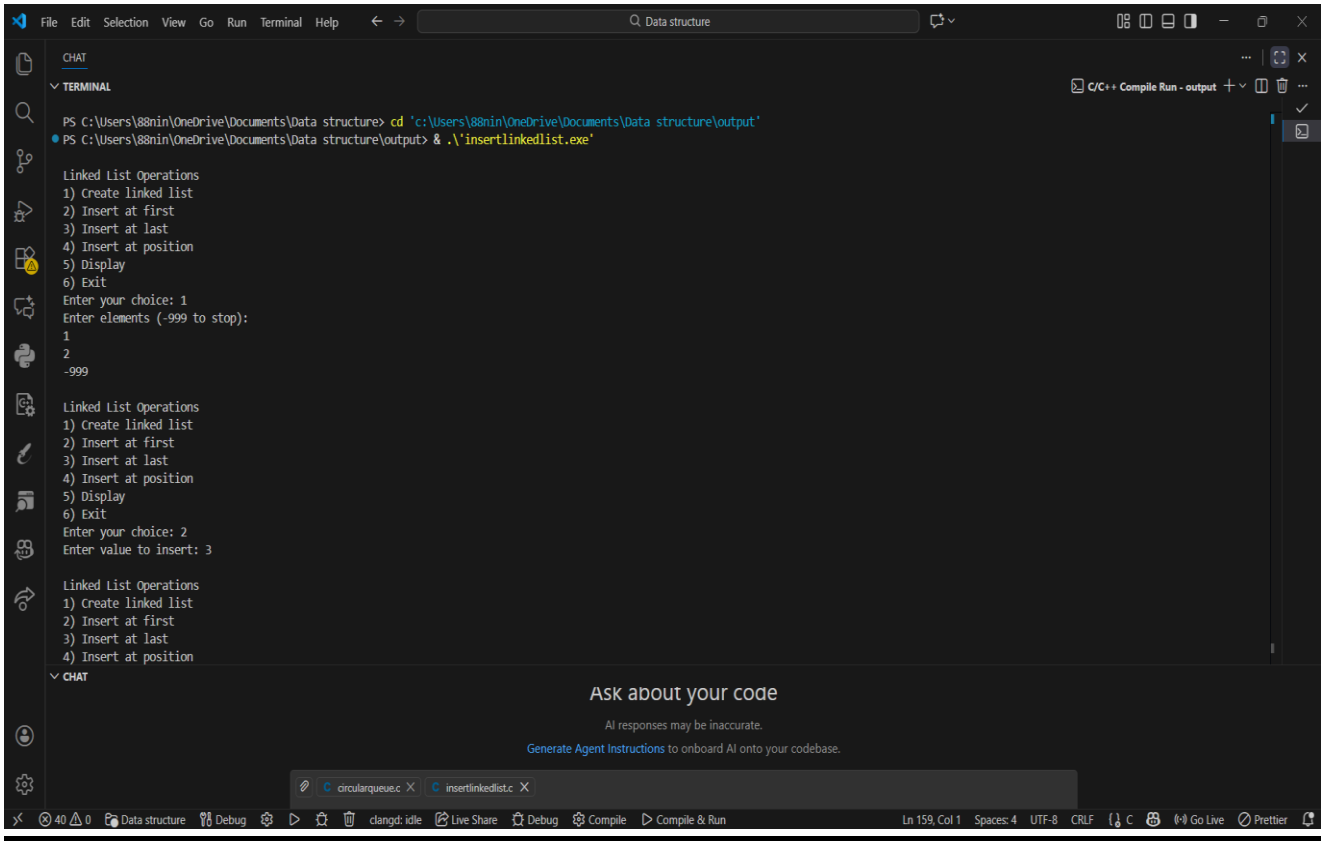
```
    while (temp != NULL) {
        if (size == pos - 1) {
            p->next = temp->next;
            temp->next = p;
            return start;
        }
        temp = temp->next;
        size++;
    }

    printf("Position out of range!\n");
    free(p);
    return start;
}

void displaylk(struct node* start) {
    struct node* temp;

    if (start == NULL) {
        printf("Linked list is empty\n");
        return;
    }

    temp = start;
```

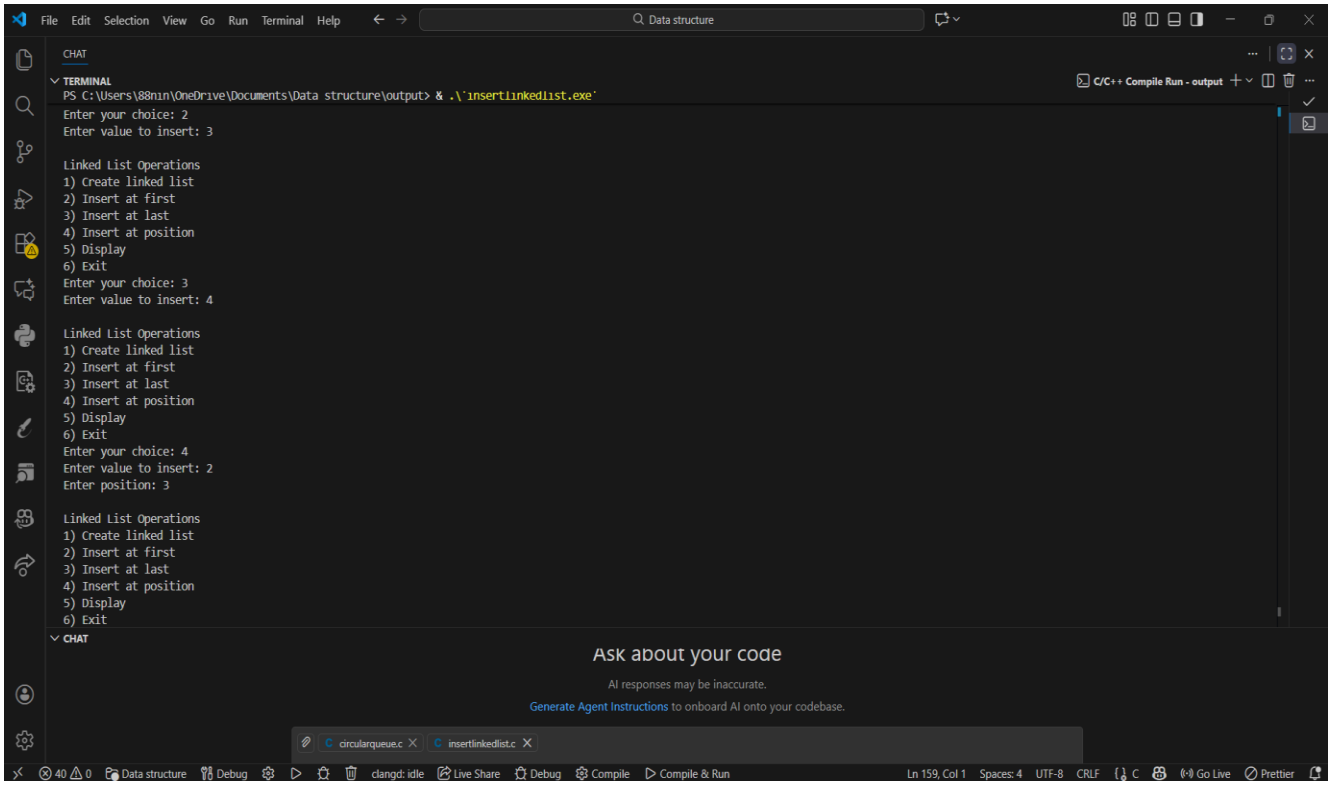
```
    printf("Elements are:\n");
    while (temp != NULL) {
        printf("%d\n", temp->info);
        temp = temp->next;
    }
}

int main() {
    struct node* head = NULL;
    int choice, val, pos;

    while (1) {
        printf("\nLinked List Operations\n");
        printf("1) Create linked list\n");
        printf("2) Insert at first\n");
        printf("3) Insert at last\n");
        printf("4) Insert at position\n");
        printf("5) Display\n");
        printf("6) Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                head = createlk();
```

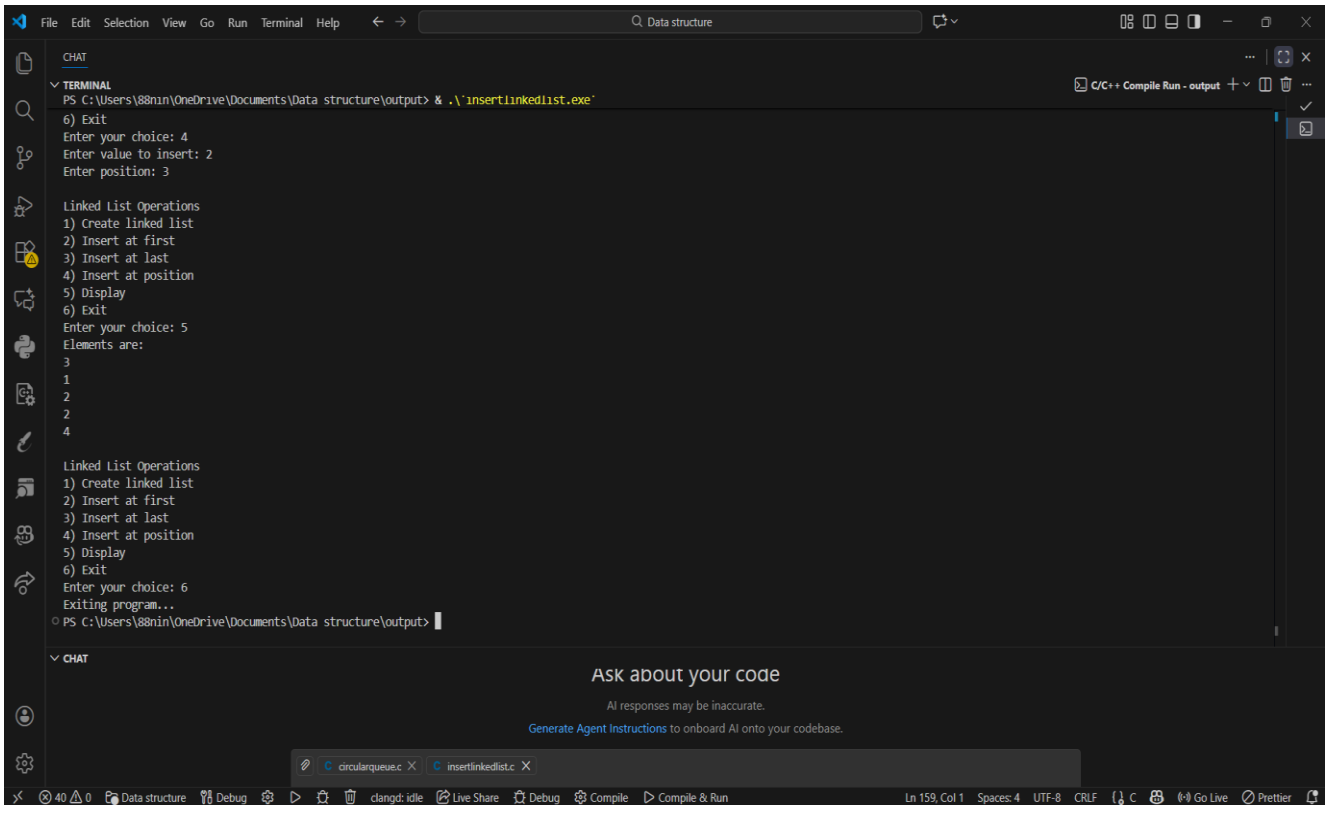
```
    break;

case 2:
    printf("Enter value to insert: ");
    scanf("%d", &val);
    head = inserfirst(head, val);
    break;

case 3:
    printf("Enter value to insert: ");
    scanf("%d", &val);
    head = insertlast(head, val);
    break;

case 4:
    printf("Enter value to insert: ");
    scanf("%d", &val);
    printf("Enter position: ");
    scanf("%d", &pos);
    head = insertatpositon(head, val, pos);
    break;

case 5:
    displaylk(head);
    break;
```

```
        case 6:
            printf("Exiting program...\n");
            return 0;

        default:
            printf("Invalid choice!\n");
    }
  }
}
```

OUTPUT:-




File Edit Selection View Go Run Terminal Help
Data structure
CHAT
TERMINAL
C/C++ Compile Run - output
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\'insertlinkedlist.exe'
Enter your choice: 2
Enter value to insert: 3
Linked List Operations
1) Create linked list
2) Insert at first
3) Insert at last
4) Insert at position
5) Display
6) Exit
Enter your choice: 3
Enter value to insert: 4
Linked List Operations
1) Create linked list
2) Insert at first
3) Insert at last
4) Insert at position
5) Display
6) Exit
Enter your choice: 4
Enter value to insert: 2
Enter position: 3
Linked List Operations
1) Create linked list
2) Insert at first
3) Insert at last
4) Insert at position
5) Display
6) Exit
CHAT
Ask about your code
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
circularqueue.c
insertlinkedlist.c
Data structure
Debug
clangd: idle
Live Share
Debug
Compile
Compile & Run
Ln 159, Col 1
Spaces: 4
UTF-8
CRLF
C
Go Live
Prettier


File Edit Selection View Go Run Terminal Help
Data structure
CHAT
TERMINAL
C/C++ Compile Run - output
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\'insertlinkedlist.exe'
6) Exit
Enter your choice: 4
Enter value to insert: 2
Enter position: 3
Linked List Operations
1) Create linked list
2) Insert at first
3) Insert at last
4) Insert at position
5) Display
6) Exit
Enter your choice: 5
Elements are:
3
1
2
2
4
Linked List Operations
1) Create linked list
2) Insert at first
3) Insert at last
4) Insert at position
5) Display
6) Exit
Enter your choice: 6
Exiting program...
PS C:\Users\88nin\OneDrive\Documents\Data structure\output>
CHAT
Ask about your code
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
circularqueue.c
insertlinkedlist.c
Data structure
Debug
clangd: idle
Live Share
Debug
Compile
Compile & Run
Ln 159, Col 1
Spaces: 4
UTF-8
CRLF
C
Go Live
Prettier