Week 5

on

Circular Linked List

Objective: In this lab, you will be working on problems related toa circular linked list. The objective is to create a circular single and double linked list and perform various operations such as **Insertion, Deletion,Traversal (displaying)**. You are required to implement the solutions using C language for the below questions.

Questions to be worked on in this lab:

1. Write a C program to perform the following operations on circularsingle linked List:
   - **a) Insertion:** Create separate functions to **insert node at the beginning, middle, and the end of the circular single linked list.**
   - **b) Traverse:** Create a separate function to display the content of each node of the circular single linked list.

Solution:

Program1.c // This is the name of the c programming file

```
#include<stdio.h>
#include <stdlib.h>
struct node {
  int data;
  struct node *next;
};
struct node *head = NULL;
// Create a new node
struct node* CreateNode() {
  struct node *newNode = (struct node *)malloc(sizeof(struct node));
  if (newNode == NULL) {
    printf("Out of memory space\n");
    exit(0);
  }
  printf("Enter the value to be inserted: ");
  scanf("%d", &newNode->data);
  newNode->next = NULL;
  return newNode;
}
```

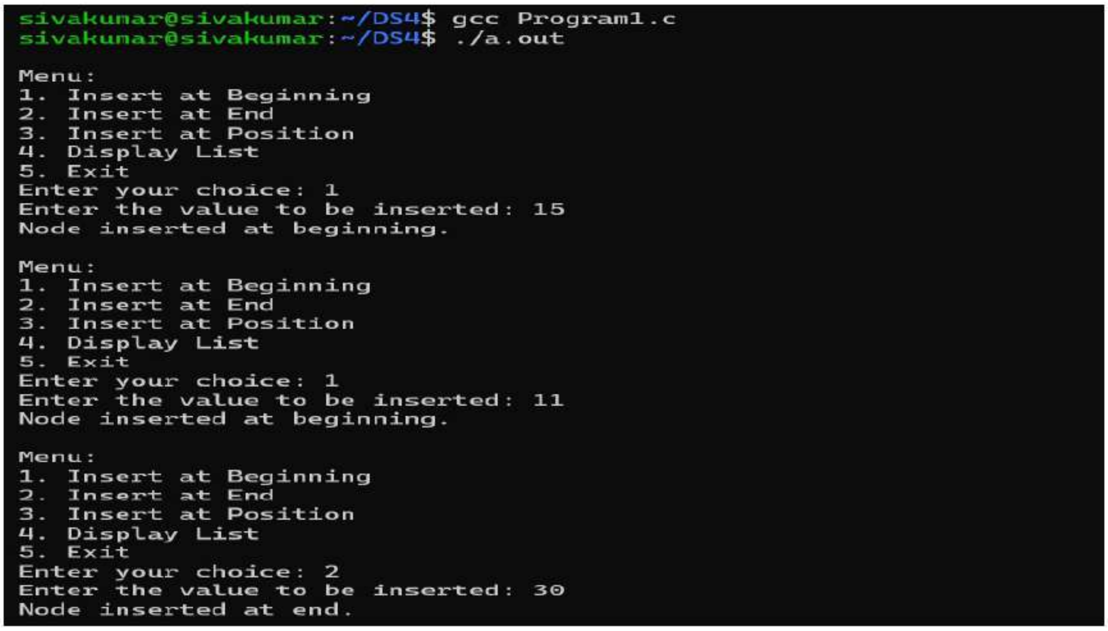
```
// Insert at beginning
void InsertAtBeginning() {
    struct node *newNode = CreateNode();
    if (head == NULL) {
        head = newNode;
        newNode->next = head;
    } else {
        struct node *temp = head;
        while (temp->next != head)
            temp = temp->next;
        newNode->next = head;
        temp->next = newNode;
        head = newNode;
    }
    printf("Node inserted at beginning.\n");
}
// Insert at end
void InsertAtEnd() {
    struct node *newNode = CreateNode();
    if (head == NULL) {
        head = newNode;
        newNode->next = head;
    } else {
        struct node *temp = head;
        while (temp->next != head)
            temp = temp->next;
        temp->next = newNode;
        newNode->next = head;
    }
    printf("Node inserted at end.\n");
}
// Insert at specific position
void InsertAtPosition(int pos) {
    int count = 1;
    struct node *newNode = CreateNode();
    if (pos == 1) {
        InsertAtBeginning();
        return;
    }
    struct node *temp = head;
    while (count < pos - 1 && temp->next != head) {
        temp = temp->next;
        count++;
    }
    if (count == pos - 1) {
        newNode->next = temp->next;
        temp->next = newNode;
        printf("Node inserted at position %d.\n", pos);
    } else {
        printf("Invalid position.\n");
        free(newNode);
    }
}
// Display the list
void DisplayList() {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    struct node *temp = head;
    printf("Circular Linked List contents:\n");
    do {
        printf(" %d ->", temp->data);
        temp = temp->next;
    } while (temp != head);
    printf(" (back to head)\n");
}
// Main function
int main() {
    int choice, pos;
    do {
        printf("\nMenu:\n");
        printf("1. Insert at Beginning\n");
        printf("2. Insert at End\n");
        printf("3. Insert at Position\n");
        printf("4. Display List\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
               InsertAtBeginning();
               break;
            case 2:
               InsertAtEnd();
               break;
            case 3:
               printf("Enter position to insert: ");
               scanf("%d", &pos);
               InsertAtPosition(pos);
               break;
            case 4:
               DisplayList();
               break;
            case 5:
               exit(0);
            default:
               printf("Invalid choice.\n");
        }
    } while (1);
    return 0;
}
```

To Compile the above code, type below command as shown:

```
gccProgram1.c
```

To see the output,execute the following as shown:

```
./a.out
```

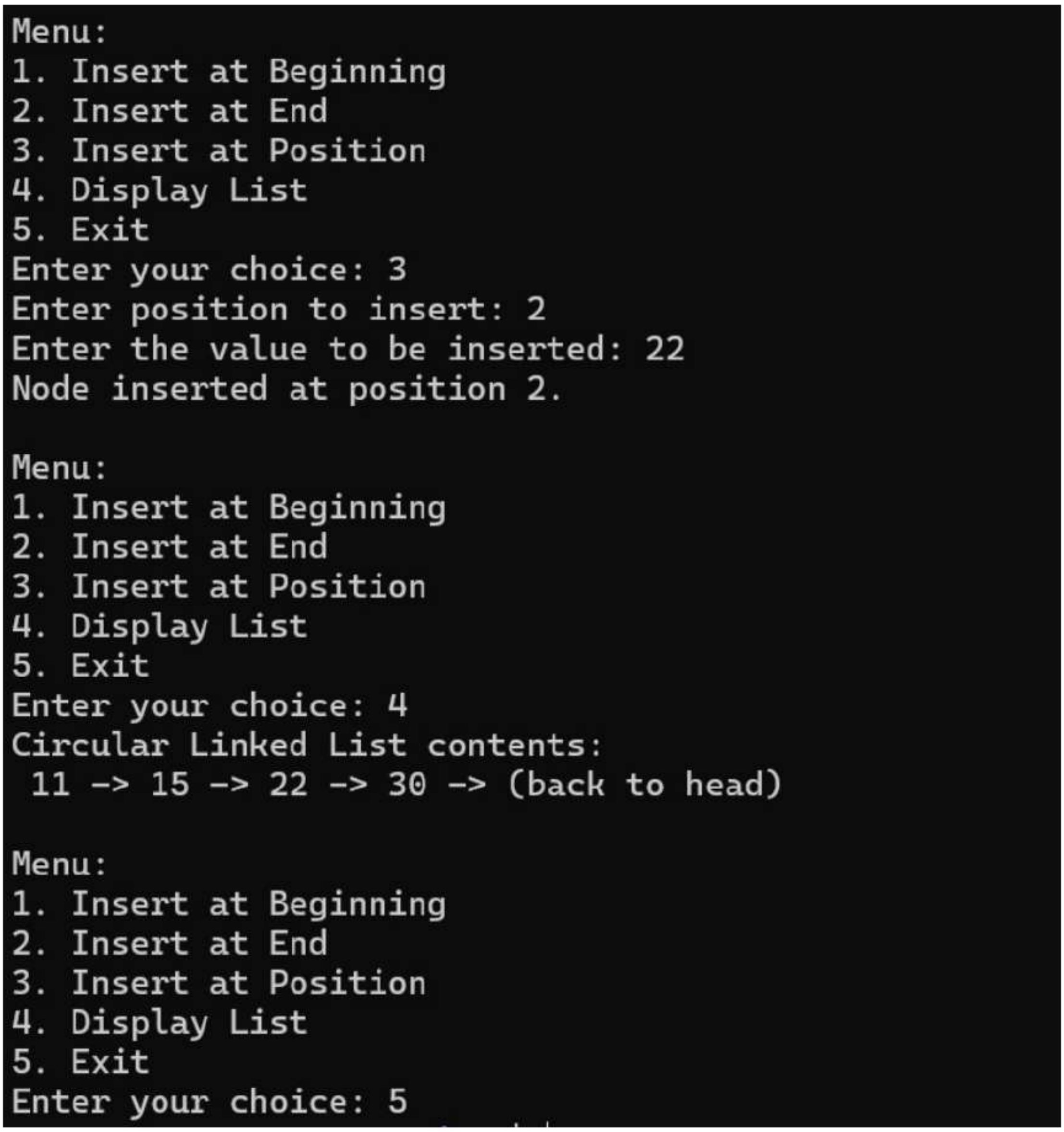


```
sivakumar@sivakumar:~/DS4$ gcc Program1.c
sivakumar@sivakumar:~/DS4$ ./a.out

Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 1
Enter the value to be inserted: 15
Node inserted at beginning.

Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 1
Enter the value to be inserted: 11
Node inserted at beginning.

Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 2
Enter the value to be inserted: 30
Node inserted at end.
```



```
Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 3
Enter position to insert: 2
Enter the value to be inserted: 22
Node inserted at position 2.

Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 4
Circular Linked List contents:
 11 -> 15 -> 22 -> 30 -> (back to head)

Menu:
1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Display List
5. Exit
Enter your choice: 5
```

2. Write a C program to perform the following operations on double linked List. The below C Program is provided a insert function to insert as many nodes as the user wants. It also provided traverse function and the main function. Your task is to add a delete function to allow a user to delete a node from the beginning, middle and the end.

Solution:

Program2.c // This is the name of the c programming file

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

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

struct node *head = NULL;

// Display the circular linked list
void DisplayList() {
  struct node *temp = head;
  if (head == NULL) {
printf("List is empty.\n");
    return;
  }

  struct node *start = head;
  do {
printf(" %d\n", temp->data);
    temp = temp->next;
  } while (temp != start);
}

// Create a circular linked list
void CreateList() {
  char ch;
```

```
   int i = 0;
   struct node *nn, *last;

nn = (struct node *)malloc(sizeof(struct node));
   if (nn == NULL) {
printf("No memory exists\n");
exit(0);
   }

printf("Enter the value for node %d: ", i);
scanf("%d", &nn->data);
nn->next = nn;
   head = nn;
   last = nn;
i++;

printf("\nPress 'n' to quit or any other key to continue: ");
fflush(stdin);
ch = getchar();

   while (ch != 'n') {
nn = (struct node *)malloc(sizeof(struct node));
      if (nn == NULL) {
printf("No memory exists\n");
exit(0);
      }

printf("Enter the value for node %d: ", i);
scanf("%d", &nn->data);
nn->next = head;
      last->next = nn;
      last = nn;
i++;

printf("\nPress 'n' to quit or any other key to continue: ");
```

```
fflush(stdin);
ch = getchar();
   }
}

// Delete from beginning
void DeleteBeginning() {
}

// Delete from end
void DeleteEnd() {
}

// Delete by position (middle)
void DeleteByPosition(int pos) {
}

int main() {
   int choice, val;
CreateList();

   do {
printf("\nMenu:\n");
printf("1. Display List\n");
printf("2. Delete from Beginning\n");
printf("3. Delete from End\n");
printf("4. Delete by Position\n");
printf("5. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

      switch (choice) {
         case 1:
DisplayList();
            break;
         case 2:
DeleteBeginning();
            break;
         case 3:
DeleteEnd();
            break;
         case 4:
printf("Enter position to delete: ");
scanf("%d", &val);
DeleteByPosition(val);
            break;
         case 5:
exit(0);
         default:
printf("Invalid choice.\n");
      }
   } while (1);

   return 0;
}
```

To Compile the above code, type below command as shown:

```
gccProgram1.c
```

To see the output,execute the following as shown after filling the code for deleting functions:

```
./a.out
```

Homework Problem for this week:

Note: Roll no and name is must for your solution sheet.

1. Write a C program to perform the following operations on a circular double linked List:
   a) **Insertion:** Create separate functions to insert as many nodes as user wants and assign data (entered by a user) to these nodes.

b) **Traverse:** Create a separate function to display the content of each node of the linked list.

c) **Search:** Create separate function to search for a given value in the nodes of the list and display its position if it is found and else "Not Found".

Test Case:

```
1: Insert
2: Search
3: Traverse
4: Quit

Enter your choice: 1
Enter value: 55
Press 'n' to quit or any other key to continue: y
Enter value: 44
Press 'n' to quit or any other key to continue: y
Enter value: 66
Press 'n' to quit or any other key to continue: y
Enter value: 33
Press 'n' to quit or any other key to continue: n

1: Insert
2: Search
3: Traverse
4: Quit

Enter your choice: 3
Values of nodes in the linked list:
55   44   66   33

Enter your choice: 2
Enter value to search: 66
Value 66 found at position 3.

Enter your choice: 4
```

Note: Support each answer with an appropriate screenshot from your system. The above homework assignment will be evaluated for 2 marks.

Submission instructions and Deadline: Submit the source code along with the screenshots of output for the above two questions respectively in a single pdf document on G-Learn. The deadline for submission is **given on G-Learn.**