Practical -7

Pra7 Implement a Circular Single Linked List (CSLL) and perform the operations: Create, Traverse, Insert_Beg, Insert_End, Delete_beg, Delete_end using Menu Driver Program.

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

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

struct node *s;

void create() {
   struct node *p, *q;
   int ch;
   p = (struct node *)malloc(sizeof(struct node));
   printf("Enter the data of the first node\n");
   scanf("%d", &p->data);
   s = p;
   do {
      q = (struct node *)malloc(sizeof(struct node));
      printf("Enter the data of the next node\n");
      scanf("%d", &q->data);
      p->next = q;
      p = q;
```

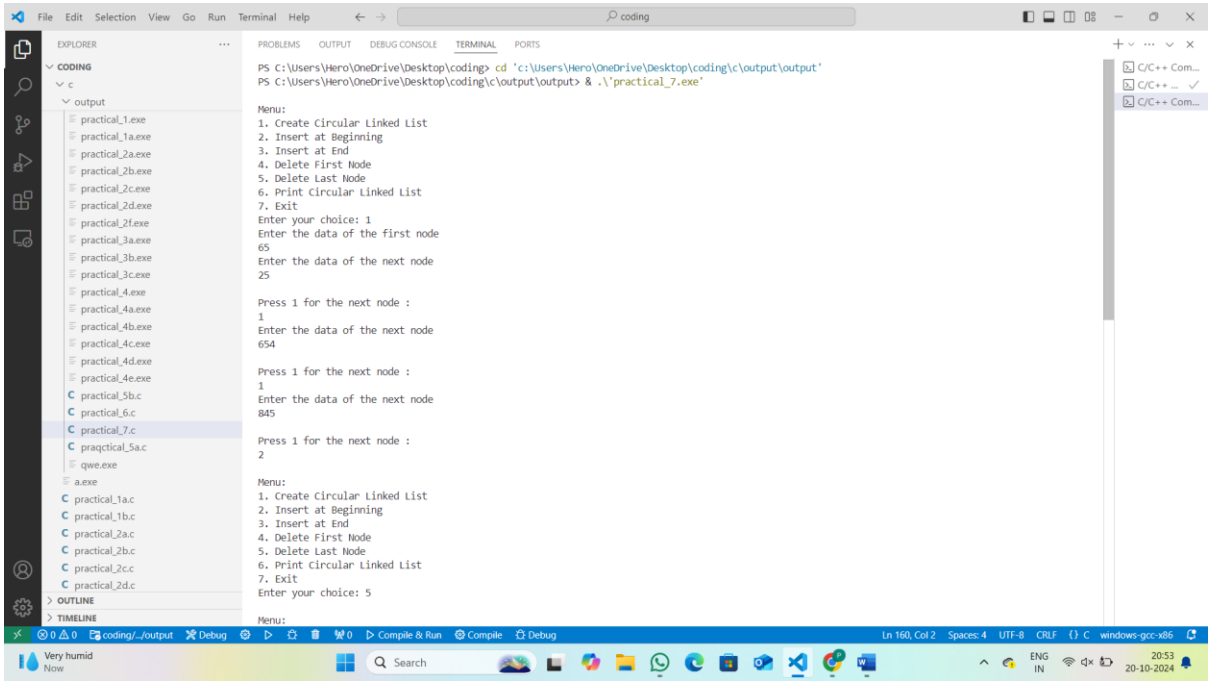
```
        printf("\nPress 1 for the next node :\n");
        scanf("%d", &ch);
    } while (ch == 1);
    p->next = s;
}

void insert_beg() {
    struct node *x, *p;
    x = (struct node *)malloc(sizeof(struct node));
    printf("Enter the data of new node\n");
    scanf("%d", &x->data);
    if (s == NULL) {
        x->next = x;
        s = x;
    } else {
        p = s;
        while (p->next != s) {
            p = p->next;
        }
        x->next = s;
        p->next = x;
        s = x;
    }
}

void insert_end() {
    struct node *x, *p;
    x = (struct node *)malloc(sizeof(struct node));
    printf("Enter the data of new node\n");
    scanf("%d", &x->data);
    if (s == NULL) {
```

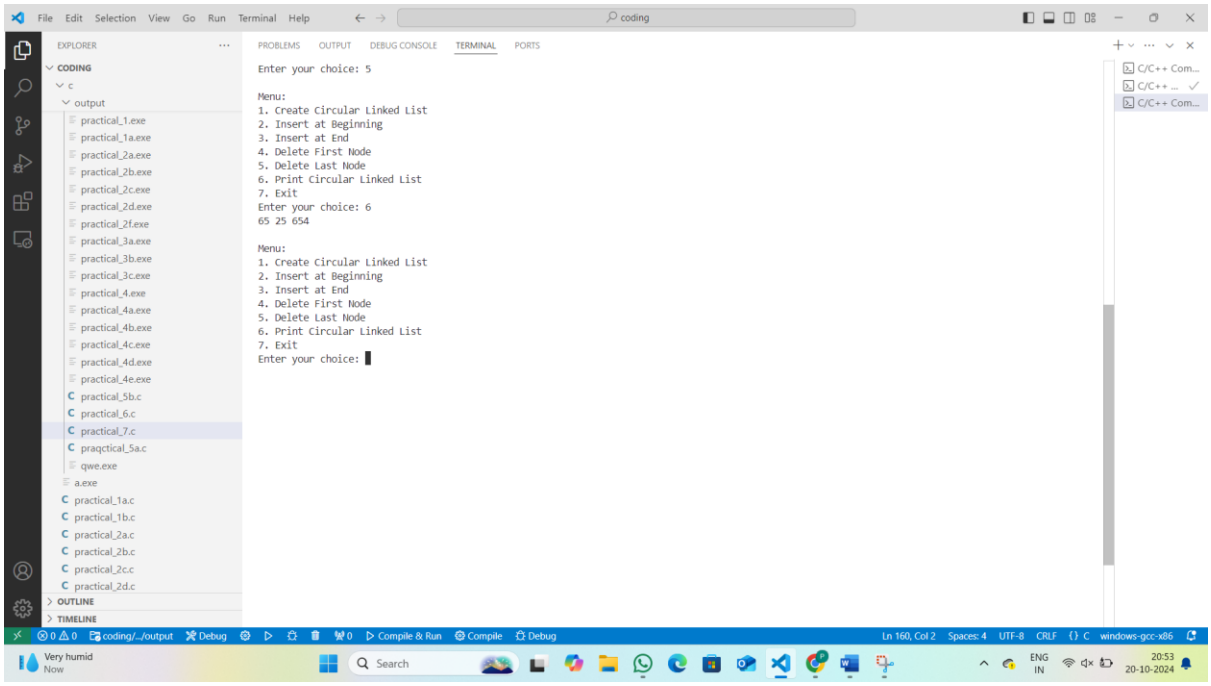
```
        x->next = x;
        s = x;
    } else {
        p = s;
        while (p->next != s) {
            p = p->next;
        }
        x->next = s;
        p->next = x;
    }
}

void delete_first() {
    struct node *q;
    if (s == NULL) {
        printf("Circular linked list is empty. Cannot delete.\n");
        return;
    }
    q = s;
    while (q->next != s) {
        q = q->next;
    }
    if (q == s) {
        free(s);
        s = NULL;
    } else {
        q->next = s->next;
        free(s);
        s = q->next;
    }
}
```

```
void delete_last() {
    struct node *p = s, *q = NULL;
    if (s == NULL) {
        printf("Circular linked list is empty. Cannot delete.\n");
        return;
    }
    while (p->next != s) {
        q = p;
        p = p->next;
    }
    if (q == NULL) {
        free(p);
        s = NULL;
    } else {
        q->next = s;
        free(p);
    }
}

void printCircularList() {
    if (s == NULL) {
        printf("Circular linked list is empty.\n");
        return;
    }
    struct node *p = s;
    do {
        printf("%d ", p->data);
        p = p->next;
    } while (p != s);
    printf("\n");
```

```
}

int main() {
    int choice;
    do {
        printf("\nMenu:\n");
        printf("1. Create Circular Linked List\n");
        printf("2. Insert at Beginning\n");
        printf("3. Insert at End\n");
        printf("4. Delete First Node\n");
        printf("5. Delete Last Node\n");
        printf("6. Print Circular Linked List\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                create();
                break;
            case 2:
                insert_beg();
                break;
            case 3:
                insert_end();
                break;
            case 4:
                delete_first();
                break;
            case 5:
                delete_last();
                break;
```

```
            case 6:
                printCircularList();
                break;
            case 7:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 7);

    return 0;
}
```




File Edit Selection View Go Run Terminal Help
coding
EXPLORER
CODING
c
output
practical_1.exe
practical_1a.exe
practical_2a.exe
practical_2b.exe
practical_2c.exe
practical_2d.exe
practical_2f.exe
practical_3a.exe
practical_3b.exe
practical_3c.exe
practical_4.exe
practical_4a.exe
practical_4b.exe
practical_4c.exe
practical_4d.exe
practical_4e.exe
practical_5b.c
practical_6.c
practical_7.c
praqctical_5a.c
qwe.exe
a.exe
practical_1a.c
practical_1b.c
practical_2a.c
practical_2b.c
practical_2c.c
practical_2d.c
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter your choice: 5
Menu:
1. Create Circular Linked List
2. Insert at Beginning
3. Insert at End
4. Delete First Node
5. Delete Last Node
6. Print Circular Linked List
7. Exit
Enter your choice: 6
65 25 654
Menu:
1. Create Circular Linked List
2. Insert at Beginning
3. Insert at End
4. Delete First Node
5. Delete Last Node
6. Print Circular Linked List
7. Exit
Enter your choice:
C/C++ Com...
C/C++ ...
C/C++ Com...
Compile & Run
Compile
Debug
Ln 160, Col 2
Spaces: 4
UTF-8
CRLF
C
windows-gcc-x86
Very humid
Now
Search
ENG
IN
20:53
20-10-2024