**PRACTICAL NO :07**

**Aim :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.**

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");

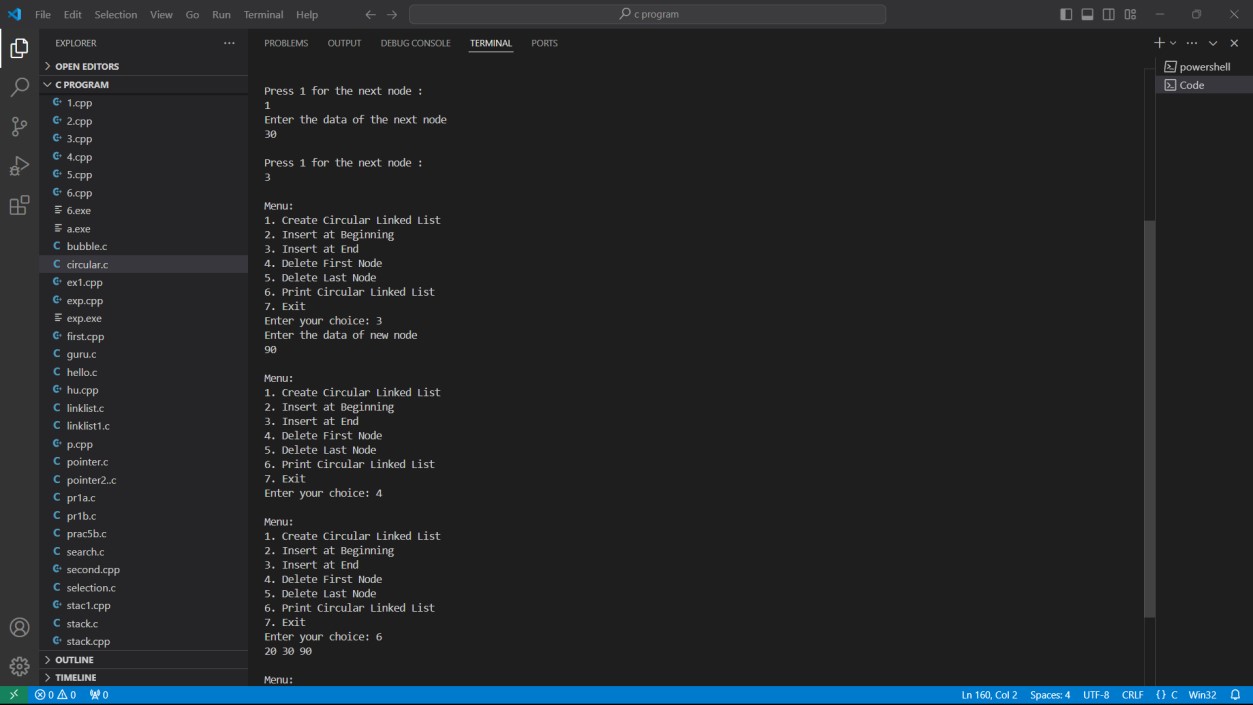
}

} while (choice != 7);

return 0;

}

Output:



Github link <https://github.com/Manas1597/DSA>