Double Circular Queue Document

ไฟล์ทั้งหมดในการทำโปรแกรม

app2.c -> ใช้ในการ **run app** เป็นการเรียกใช้ **2** ไฟล์ข้างล่าง

DoubleCircularQueue.c -> เป็น **function** กระบวนการทั้งหมด ของ **DCQ**

System.c -> ใช้ในการแสดงผล และ ดัก **input** ต่างๆ

ช่องทางเพิ่มเติม หากต้องการนำไปใช้งาน

Github : NutNaphop/Circular_Array_C: Using for Project (github.com)

app2.c

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include "DoubleCircularQueue.c"
#include "System.c"

int isRunning = 1;
int val;
char input[2];

int main() {
    while (isRunning) {
        clear() ;
        displayMenu();
        printf("Enter your choice: ");
        scanf("%s", input);
        while (getchar() != '\n') ; // Using for get input only one word , with out whitespace
        clear() ;
        if (isInt(input)) {
            switch (input[0]) {
                case '1' :
                    displayEF() ;
                    printf("Enter the number you want to enqueue at the front: ");
                    scanf("%s", input);
                    if (isInt(input)) {
                        val = atoi(input);
                        enQueueFront(val);
                    }
                    else {
                        printf("Invalid input. Please enter a valid integer.\n");
                    }
                    break;
                case '2':
                    displayER() ;
                    printf("Enter the number you want to enqueue at the rear: ");
                    scanf("%s", input);
                    if (isInt(input)){
                        val = atoi(input);
                        enQueueRear(val);
                    }
                    else {
                        printf("Invalid input. Please enter a valid integer.\n");
                    }
                    break;
                case '3':
                    displayDF() ;
                    deQueueFront();
                    break;

                case '4':
                    displayDR() ;
                    deQueueRear();
                    break;

                case '5':
                    displayItem() ;
                    showItem() ;
                    break;
                case '6':
                    displayEXIT() ;
                    isRunning = 0;
                    printf("Exiting the program.\n");
                    break;

                default:
                    printf("Invalid choice. Please try again.\n");
                    break;
            }
            wait() ;
        }
        else {
            printf("Invalid input. Please enter a valid integer.\n");
            wait() ;
        }
    }
}
```

DoubleCircularQueue.c

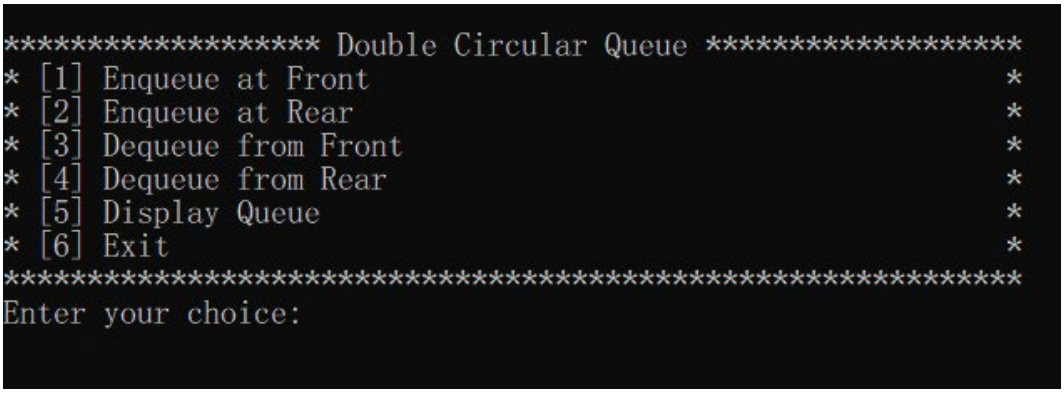
```c
#include <stdio.h>
#include <stdlib.h>

#define SIZE 5
int items[SIZE];
int front = -1, rear = -1;

// Function
int enQueueFront(int data);
int enQueueRear(int data);
int deQueueFront();
int deQueueRear();
void showItem();
int isFull();
int isEmpty();


int enQueueFront(int data) {
    if (isFull()) {
        printf("Queue is full. Cannot enqueue at the front.\n");
        return 0;
    }
    if (isEmpty()) {
        front = 0;
        rear = 0;
    }
    else {
        front = (front - 1 + SIZE) % SIZE;
    }
    items[front] = data;
    printf("Enqueued %d at the front. Enqueue Front Complete.\n", items[front]);
    showItem() ;
}

int enQueueRear(int data) {
    if (isFull()) {
        printf("Queue is full. Cannot enqueue at the rear.\n");
        return 0;
    }
    if (isEmpty()) {
        front = 0;
        rear = 0;
    }
    else {
        rear = (rear + 1) % SIZE;
    }
    items[rear] = data;
    printf("Enqueued %d at the rear. Enqueue Complete.\n", items[rear]);
    showItem() ;
}

int deQueueFront() {
    if (isEmpty()) {
        printf("Queue is empty. Cannot dequeue from the front.\n");
        return 0;
    }
    printf("Dequeued item: %d. Dequeue Complete!\n", items[front]);
    if (front == rear) {
        front = -1;
        rear = -1;
    }
    else {
        front = (front + 1) % SIZE;
    }
    showItem() ;
}

int deQueueRear() {
    if (isEmpty()) {
        printf("Queue is empty. Cannot dequeue from the rear.\n");
        return 0;
    }
    printf("Dequeued item: %d. Dequeue Complete!\n", items[rear]);
    if (front == rear) {
        front = -1;
        rear = -1;
    }
    else {
        rear = (rear - 1 + SIZE) % SIZE;
    }
    showItem() ;
}

void showItem() {
    printf("*****************************************************************\n");
    printf("Front -> %d\n", front);

    if (isEmpty()) {
        printf("Queue is empty. Nothing to display.\n");
    }

    else {
        printf("\n") ;
        for (int i = front; i != rear; i = (i + 1) % SIZE) {
            printf("%d ->", items[i]);
        }
        printf("%d" , items[rear]) ;
    }
    printf("\n\nRear -> %d\n", rear);
    printf("*****************************************************************\n");
}

int isFull() {
    if ((rear + 1) % SIZE == front) {
        return 1;
    }
    return 0;
}

int isEmpty() {
    if (front == -1 && rear == -1) {
        return 1;
    }
    return 0;
}
```

System.c

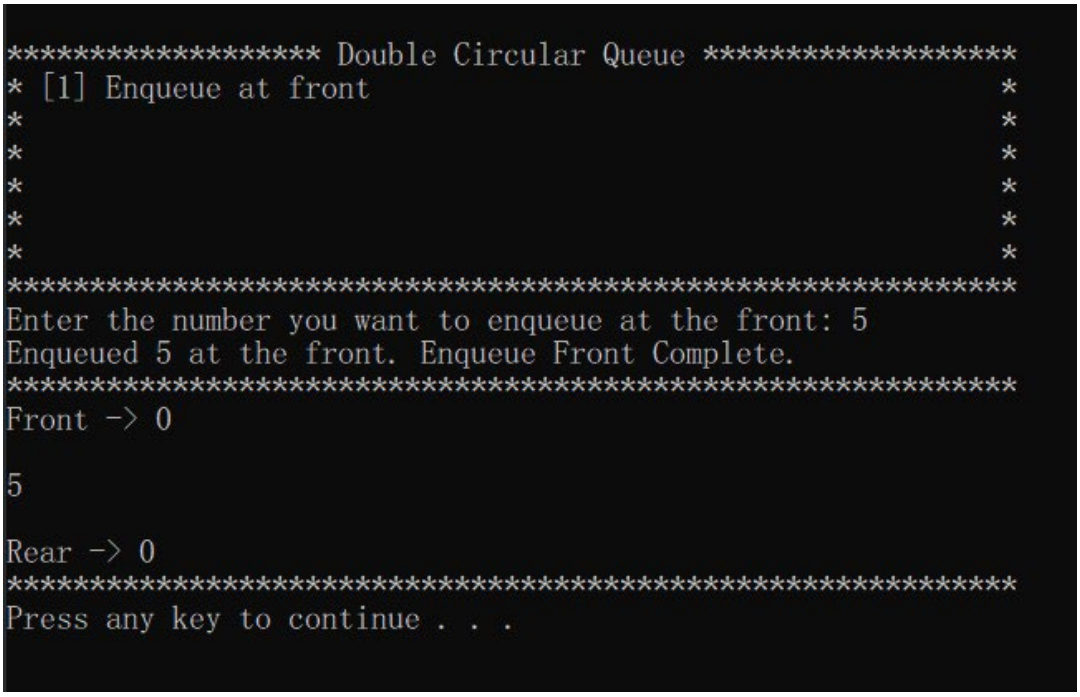
```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

void clear();
void wait();
int isInt(char *str) ;
void clear() ;
void wait() ;

void displayMenu() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("* [1] Enqueue at Front                                    *\n");
    printf("* [2] Enqueue at Rear                                     *\n");
    printf("* [3] Dequeue from Front                                  *\n");
    printf("* [4] Dequeue from Rear                                   *\n");
    printf("* [5] Display Queue                                       *\n");
    printf("* [6] Exit                                                *\n");
    printf("***********************************************************\n");
}

void displayEF() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("* [1] Enqueue at front                                    *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("***********************************************************\n");
}

void displayER() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("*                                                         *\n");
    printf("* [2] Enqueue at Rear                                     *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("***********************************************************\n");
}

void displayDF() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("* [3] Dequeue from Front                                  *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("***********************************************************\n");
}

void displayDR() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("* [4] Dequeue from Rear                                   *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("***********************************************************\n");
}

void displayItem() {
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("* [5] Display Queue                                       *\n");
    printf("*                                                         *\n");
    printf("***********************************************************\n");
}

void displayEXIT(){
    printf("\n");
    printf("****************** Double Circular Queue ******************\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("*                                                         *\n");
    printf("* [6] Exit                                                *\n");
    printf("***********************************************************\n");
}

int isInt(char *str) {
    while (*str) {
        if (!isdigit(*str)) {
            return 0;
        }
        str++;
    }
    return 1;
}

void clear() {
    system("cls");
}

void wait() {
    system("pause");
}

```

Example of program display

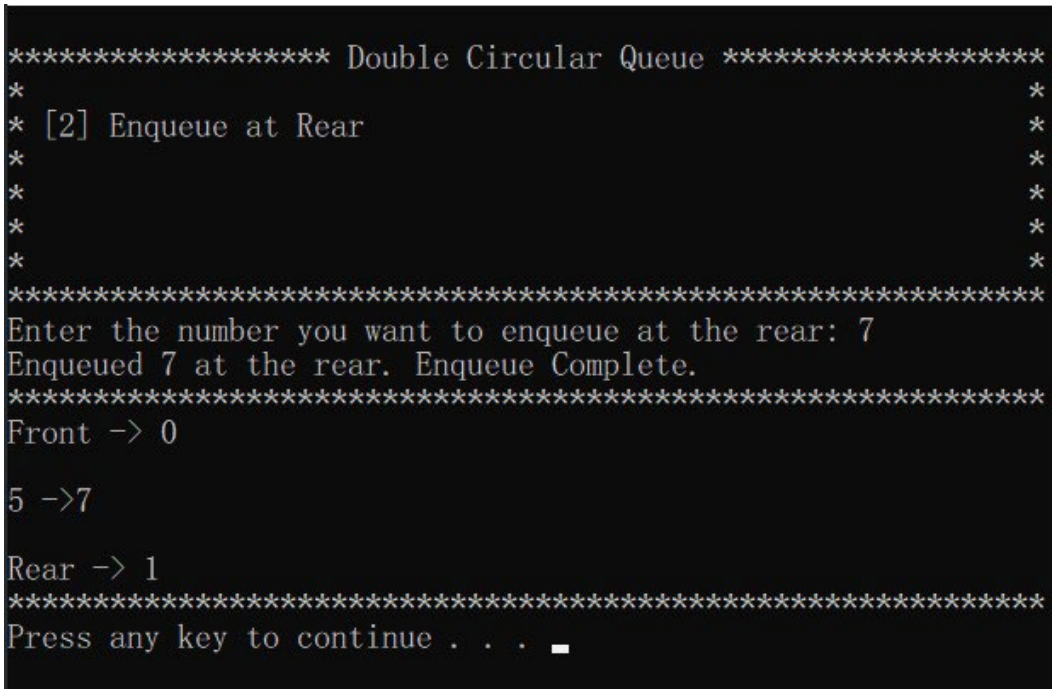
#การแสดงผลทางหน้าจอเมื่อเปิดโปรแกรม



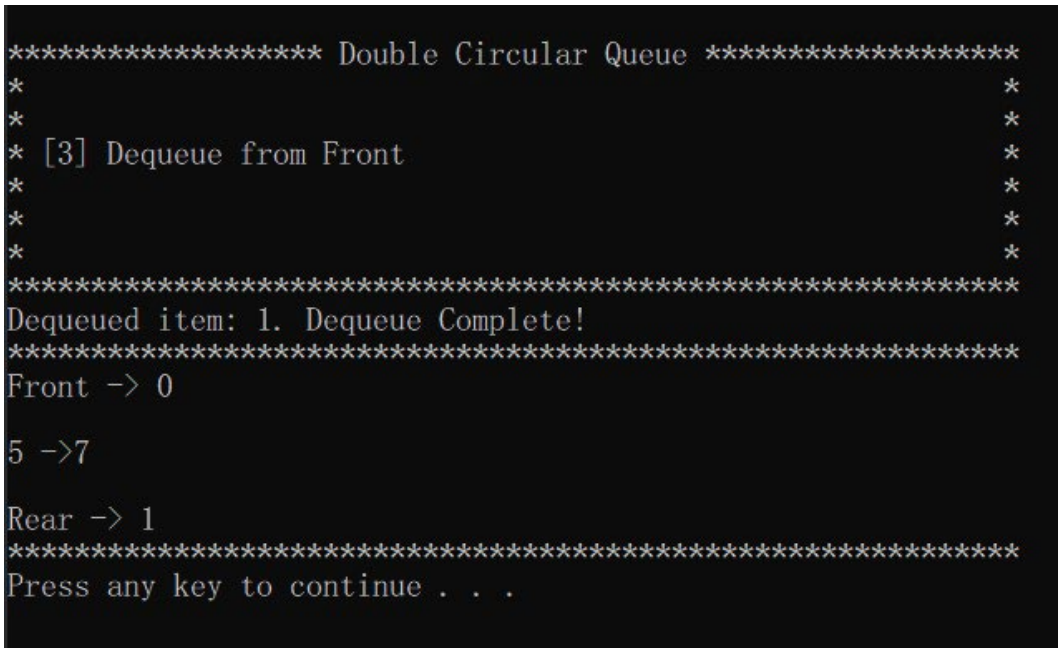
#เมื่อเลือก [1] Enqueue at Front



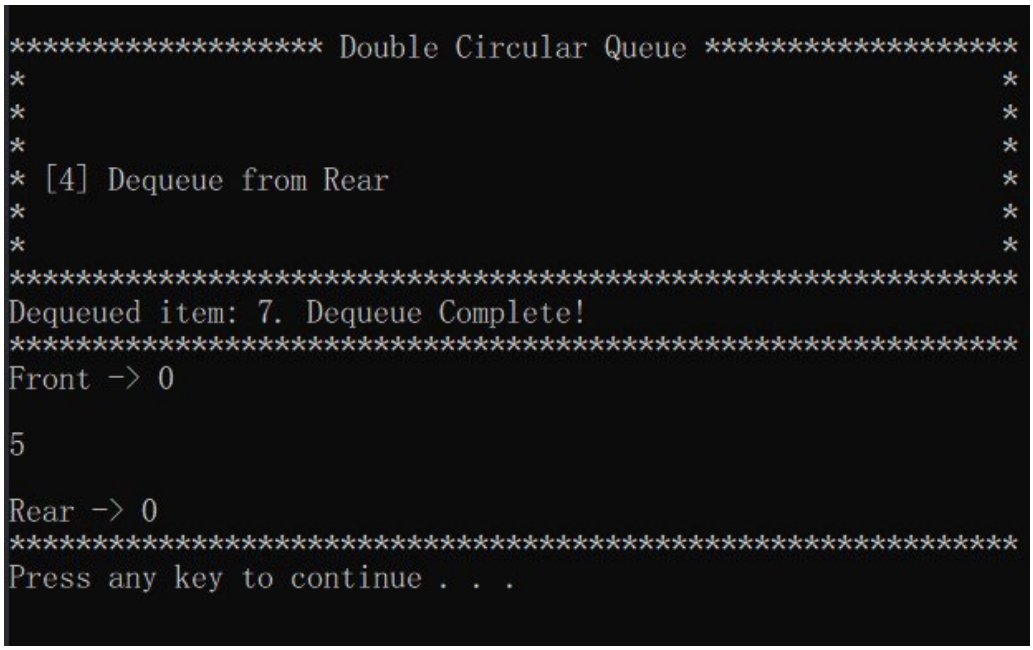
#เมื่อเลือก [2] Enqueue at Rear



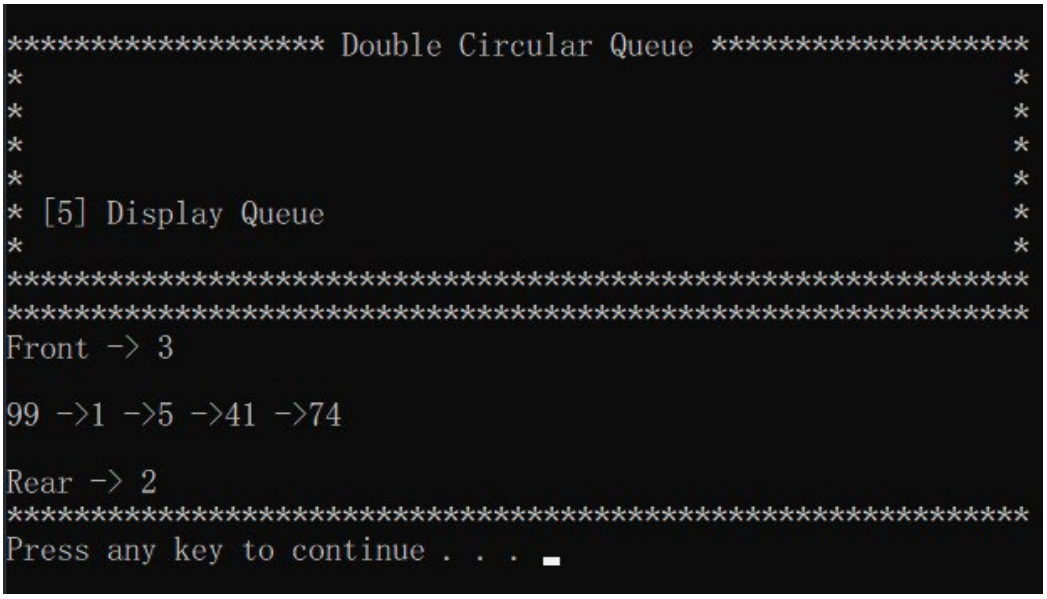
#เมื่อเลือก [3] Dequeue at Front



#เมื่อเลือก [4] Dequeue at Rear



#เมื่อเลือก [5] Display Queue



#เมื่อ Input ค่าที่ไม่ใช่ Integer