DOUBLE ENDED QUEUE

#include <stdio.h>  
  #include <stdlib.h>  
  
  struct node {  
        int data;  
        struct node \*prev, \*next;  
  };  
  
  struct node \*head = NULL, \*tail = NULL;  
  
  struct node \* createNode(int data) {  
        struct node \*newnode = (struct node \*)malloc(sizeof (struct node));  
        newnode->data = data;  
        newnode->next = newnode->prev = NULL;  
        return (newnode);  
  }  
  
   
  
  void createSentinels() {  
        head = createNode(0);  
        tail = createNode(0);  
        head->next = tail;  
        tail->prev = head;  
  }  
  
  /\* insertion at the front of the queue \*/

  void enqueueAtFront(int data) {

        struct node \*newnode, \*temp;

        newnode = createNode(data);

        temp = head->next;

        head->next = newnode;

        newnode->prev = head;

        newnode->next = temp;

        temp->prev = newnode;

  }

  /\*insertion at the rear of the queue \*/

  void enqueueAtRear(int data) {

        struct node \*newnode, \*temp;

        newnode = createNode(data);

        temp = tail->prev;

        tail->prev = newnode;

        newnode->next = tail;

        newnode->prev = temp;

        temp->next = newnode;

  }

  /\* deletion at the front of the queue \*/

  void dequeueAtFront() {

        struct node \*temp;

        if (head->next == tail) {

                printf("Queue is empty\n");

        } else {

                temp = head->next;

                head->next = temp->next;

                temp->next->prev = head;

                free(temp);

        }

        return;

  }

  /\* deletion at the rear of the queue \*/

  void dequeueAtRear()  {

        struct node \*temp;

        if (tail->prev == head) {

                printf("Queue is empty\n");

        } else {

                temp = tail->prev;

                tail->prev = temp->prev;

                temp->prev->next = tail;

                free(temp);

        }

        return;

  }

  /\* display elements present in the queue \*/

  void display() {

        struct node \*temp;

        if (head->next == tail) {

                printf("Queue is empty\n");

                return;

        }

        temp = head->next;

        while (temp != tail) {

                printf("%-3d", temp->data);

                temp = temp->next;

        }

        printf("\n");

  }

  int main() {

        int data, ch;

        createSentinels();

        while (1) {

                printf("1. Enqueue at front\n2. Enqueue at rear\n");

                printf("3. Dequeue at front\n4. Dequeue at rear\n");

                printf("5. Display\n6. Exit\n");

                printf("Enter your choice:");

                scanf("%d", &ch);

                switch (ch) {

                        case 1:

                                printf("Enter the data to insert:");

                                scanf("%d", &data);

                                enqueueAtFront(data);

                                break;

                      case 2:

                                printf("Enter ur data to insert:");

                                scanf("%d", &data);

                                enqueueAtRear(data);

                        case 3:

                                dequeueAtFront();

                                break;

                        case 4:

                                dequeueAtRear();

                                break;

                        case 5:

                                display();

                                break;

                        case 6:

                                exit(0);

                        default:

                                printf("Pls. enter correct option\n");

                                break;

                }

        }

        return 0;

  }

