5/01/24

Queue implementation using linked list

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

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

struct Node *createNode(int data) {
    struct Node *newNode = (struct Node*) malloc(sizeof(
    struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void enqueue(struct Node **front, struct Node **
rear, int data) {
    struct Node *newnode = createNode(data);
    if (*front == NULL) {
        *front = *rear = newNode;
        return;
    }
```

```
    (*rear)->next = newNode;
    *rear = newNode;
}

int dequeue(struct Node **front, struct Node **rear){
    if(*front == NULL){
        printf("Queue is empty\n");
        return -1;
    }
    int data = (*front)->data;
    struct Node *Temp = *front;
    if(*front == *rear){
        *front = *rear = NULL;
    } else {
        *front = (*front)->next;
    }
    free(Temp);
    return data;
}

void printQueue(struct Node *front){
    printf("Queue is elements:");
    while(front != NULL){
        printf("%d", front->data);
        front = front->next;
    }
    printf("\n");
}
```

```
int main() {
    struct Node* queuefront = NULL;
    struct Node* queuerear = NULL;
    int choice, data;

    do {
        printf("\n choose an operation:\n");
        printf("1. Enqueue");
        printf("2. Dequeue");
        printf("3. Print queue");
        switch(choice) {
        case 1:
            printf("Enter the element to queue into the
            queue:");
            scanf("%d", &data);
            enqueue(&queuefront, &queuerear, data);
            break;

        case 2:
            printf("Dequeued element from the queue:%d\n",
            dequeue(&queueFront, &queuerear));

        case 3:
            printfQueue(queue font);
            break;
```

```
        case 0:
            printf("Exit program.\n");
            break;
        }
    } while (choice != 0)
    return 0
}
```

Output:- choose an operation

1) Push
2) Pop
3) Print Stack
0) Exit

Enter your choice : 1
Enter the elements to push in the stack : 2

Enter the elements to push in the stack : 3

Enter the elements to push in the stack : 4

Enter your choice : 2
Popped element from stack : 4

Enter your choice: 3
stack elements: 3 2

Enter your choice: 4
Exit