

DATA STRUCTURE

Project 3

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#define QUEUE_SIZE 100
#define STACK_SIZE 100

struct Queue
{
    int items[QUEUE_SIZE];
    int front, rear;
};

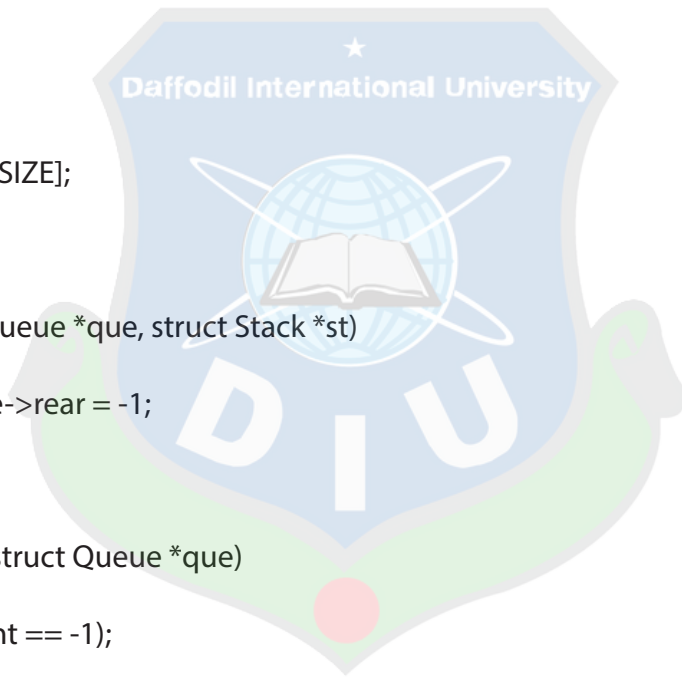
struct Stack
{
    int items[STACK_SIZE];
    int top;
};

void initial(struct Queue *que, struct Stack *st)
{
    que->front = que->rear = -1;
    st->top = -1;
}

int Queue_Empty(struct Queue *que)
{
    return (que->front == -1);
}

int Queue_Full(struct Queue *que)
{
    return (que->rear == QUEUE_SIZE - 1);
}
```

NAME: Md. Hasin Almas Mitul
ID: 213-15-4617
SECTION: 61_D
Department of CSE



```
void enqueue(struct Queue *que, int value)
{
    if (Queue_Full(que))
    {
        printf("Queue full. Enqueue is not possible.\n");
        return;
    }

    if (Queue_Empty(que))
    {
        que->front = que->rear = 0;
    }
    else
    {
        que->rear++;
    }

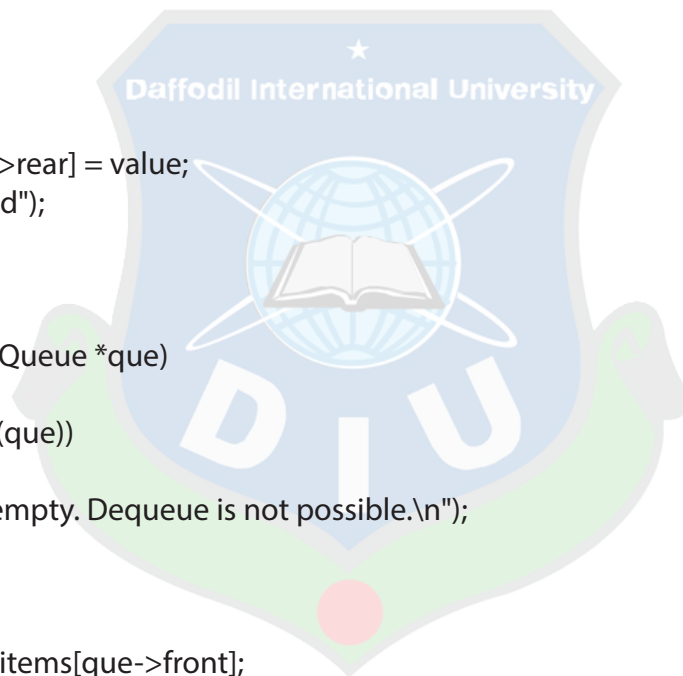
    que->items[que->rear] = value;
    printf("Item added");
}
```

```
int dequeue(struct Queue *que)
{
    if (Queue_Empty(que))
    {
        printf("Queue empty. Dequeue is not possible.\n");
        return -1;
    }
```

```
    int value = que->items[que->front];
```

```
    if (que->front == que->rear)
    {
        que->front = que->rear = -1;
    }
    else
    {
        que->front++;
    }
```

```
    return value;
}
```



```
int Stack_Empty(struct Stack *st)
{
    return (st->top == -1);
}
```

```
int Stack_Full(struct Stack *st)
{
    return (st->top == STACK_SIZE - 1);
}
```

```
void push(struct Stack *st, int value)
{
    if (Stack_Full(st))
    {
        printf("Stack full. Cannot push more item.\n");
        return;
    }
    st->items[++st->top] = value;
}
```

```
int pop(struct Stack *st)
{
    if (Stack_Empty(st))
    {
        printf("Stack empty. Cannot pop item.\n");
        return -1;
    }
    return st->items[st->top--];
}
```

```
void display_Queue(struct Queue *que)
{
    if (Queue_Empty(que))
    {
        printf("Queue is empty.\n");
        return;
    }

    printf("Queue elements: ");
    for (int i = que->front; i <= que->rear; i++)
    {
        printf("%d ", que->items[i]);
    }
    printf("\n");
}
```

```

void display_Stack(struct Stack *st)
{
    if (Stack_Empty(st))
    {
        printf("Stack is empty.\n");
        return;
    }

    printf("Stack elements: ");
    for (int i = st->top; i >= 0; i--)
    {
        printf("%d ", st->items[i]);
    }
    printf("\n");
}

```

```

int main()
{
    struct Queue que;
    struct Stack st;
    initial(&que, &st);

    int ch, ele;

    while (1)
    {
        printf("\n1. Request to Book A Ticket\n2. Cancel The Request\n
              3. Update The Request\n4. Update Available Tickets\n
              5. Display Requested Tickets\n6. Display Available Tickets\n7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &ch);
    }
}

```



```
switch (ch)
{
    case 1:
        printf("Enter the ticket request: ");
        scanf("%d", &ele);
        enqueue(&que, ele);
        break;

    case 2:
        ele = dequeue(&que);
        if (ele != -1)
        {
            printf("Canceled request for a ticket: %d\n", ele);
        }
        break;

    case 3:
        printf("Enter the updated ticket request: ");
        scanf("%d", &ele);
        enqueue(&que, ele);
        break;

    case 4:
        printf("Enter the number of available tickets to add: ");
        scanf("%d", &ele);
        for (int i = 0; i < ele; i++)
        {
            push(&st, i + 1);
        }
        break;

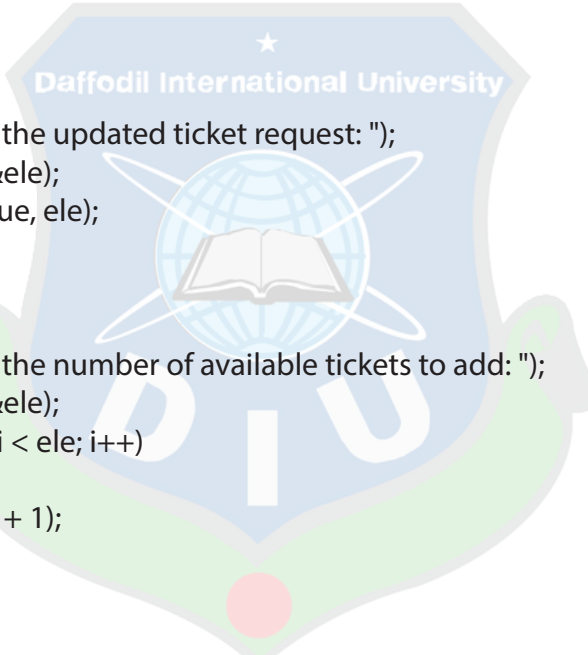
    case 5:
        display_Queue(&que);
        break;

    case 6:
        display_Stack(&st);
        break;

    case 7:
        exit(0);

    default:
        printf("Invalid choice.\n");
}

return 0;
}
```

The logo of Daffodil International University is a shield-shaped emblem. At the top, it features a five-pointed star above the university's name, "Daffodil International University", which is written in a blue serif font. The central part of the shield contains a stylized globe with latitude and longitude lines, overlaid with a white open book. The shield is flanked by two green leaves on the sides and a red circular element at the bottom center. The entire logo is rendered in a light, semi-transparent style, serving as a background watermark for the code.

OUTPUT:

```
"D:\Programming\C Program" X + v
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice: 1
Enter the ticket request: 5
Item added
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice: 1
Enter the ticket request: 56
Item added
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice: 5
Queue elements: 5 56
```

```
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice: 4
Enter the number of available tickets to add: 5
```

```
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice: 6
Stack elements: 5 4 3 2 1
```

```
1. Request to Book A Ticket
2. Cancel The Request
3. Update The Request
4. Update Available Tickets
5. Display Requested Tickets
6. Display Available Tickets
7. Exit
Enter your choice:
```