



DSA LAB

Assignment 04

Name: Madhav Jha

Roll no.: E3-48

Branch: CSE (AI & ML)

Bus Queue

A bus has a capacity of 30 passengers.

there is a single door for entry in bus and two doors for exit. Passengers can occupy seats in front or rear sections of bus on first in first out basis. the entry door automatically gets closed when bus is full and alarms for the same.

Passengers once occupied the sets are not asked to change it. Identify the best data structure that can be used and implement the program.

Code:

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

void err(char s[]) {
    printf("\nERROR: %s!!!\n", s);
}

typedef struct {
    int front;
    int rear;
    int* arr;
    int maxSize;
}Queue;

int count = 0;
Queue* initQ(int n) {
    Queue* temp = (Queue*)malloc(sizeof(Queue));
    temp->maxSize = n;
    temp->front = -1;
    temp->rear = -1;
    temp->arr = (int*)malloc(temp->maxSize * sizeof(int));
    return temp;
}

void displayBus(Queue* l, Queue* r) {
    printf("\nExit ");
    if (l->front != -1 && l->rear != -1) {
        for (int i = l->front; i < l->rear + 1; i++) {
            printf("p%d ", l->arr[i]);
        }
    }
}
```

```

    }
}

printf(" Entry ");

if (r->front != -1 && r->rear != -1) {
    for (int i = r->rear; i >= r->front; i--) {
        printf("p%d ", r->arr[i]);
    }
}
printf(" Exit\n");
}

void pushElem(Queue* q) {
    if (q->rear == q->maxSize - 1) {
        err("Overflow");
        return;
    }
    q->arr[++(q->rear)] = ++count;
    if (q->front == -1) {
        (q->front)++;
    }
}

void leftExit(Queue* l, Queue* r) {
    if (l->rear == -1 && r->rear == -1) {
        err("Underflow");
        return;
    }

    if (l->rear != -1 && l->front <= l->rear) {
        ++(l->front);
    }
    else if (r->rear != -1 && r->front <= r->rear) {
        --(r->rear);
    }
    else {
        err("Underflow");
        return;
    }
}

```

```

}

void rightExit(Queue* l, Queue* r) {
    if (l->rear == -1 && r->rear == -1) {
        err("Underflow");
        return;
    }

    if (r->rear != -1 && r->front <= r->rear) {
        ++(r->front);
    }
    else if (l->rear != -1 && l->front <= l->rear) {
        --(l->rear);
    }
    else {
        err("Underflow");
        return;
    }
}

int main(void) {
    Queue* leftBus = initQ(15);
    Queue* rightBus = initQ(15);

    int choice = 0;
    do {
        printf("\n0. Enter 0 to exit!");
        printf("\n1. Add passenger and sit in the front.");
        printf("\n2. Add passenger and sit in the back.");
        printf("\n3. Exit passenger from the front.");
        printf("\n4. Exit passenger from the back.");
        printf("\n5. Display Bus.");

        printf("\nYour choice: ");
        scanf("%d", &choice);

        int inp, src;
        switch (choice) {
            case 0:

```

```

        printf("\n\nExit...!\n\n");
        break;
    case 1:
        pushElem(leftBus);
        printf("\n\nPassenger added!!\n");
        break;
    case 2:
        pushElem(rightBus);
        printf("\n\nPassenger added!!\n");
        break;
    case 3:
        leftExit(leftBus, rightBus);
        printf("\n\nPassenger exited from front.\n");
        break;
    case 4:
        rightExit(leftBus, rightBus);
        printf("\n\nPassenger exited from front.\n");
        break;
    case 5:
        printf("\n");
        displayBus(leftBus, rightBus);
        break;
    default:
        printf("\nERROR: Invalid choice!!!");
        break;
    }
} while (choice != 0);

return 0;
}

```

Output:

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 1
```

Passenger added!!

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 1
```

Passenger added!!

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 2
```

Passenger added!!

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 2
```

Passenger added!!

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 5
```

Exit p1 p2 Entry p4 p3 Exit

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 3
```

Passenger exited from front.

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 5
```

```
Exit p2  Entry p4 p3  Exit
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 3
```

```
Passenger exited from front.
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 5
```

```
Exit  Entry p4 p3  Exit
```



```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 4
```

```
Passenger exited from front.
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 5
```

```
Exit  Entry p4  Exit
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 3
```

```
Passenger exited from front.
```

```
1. Add passenger and sit in the front.
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 5
```

```
Exit  Entry  Exit
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 3
```

```
ERROR: Underflow!!!
```

```
Passenger exited from front.
```

```
0. Enter 0 to exit!
1. Add passenger and sit in the front.
2. Add passenger and sit in the back.
3. Exit passenger from the front.
4. Exit passenger from the back.
5. Display Bus.
Your choice: 0
```

```
Exit...!
```