

```
#include <stdio.h>
#include <stdlib.h>
#define N 5

struct fila {
    int ini;
    int fim;
    int cont;
    int vet[N];
};

typedef struct fila Fila;

Fila* cria(void);
int incr(int i);
int decr(int i);
void inserir(Fila* f, int v);
void retirar(Fila* f);
void imprimir(Fila* f);
Fila* libera(Fila* f);
int quantidade(Fila* f);
void fura_fila(Fila* f, int x);

int main() {
    Fila* f1;
    f1 = cria();

    inserir(f1, 2);
    inserir(f1, 3);
    inserir(f1, 4);

    printf("Fila original: ");
    imprimir(f1);

    printf("\nQuantidade de elementos: %d\n", quantidade(f1));

    printf("Furando a fila com o numero 99...\n");
    fura_fila(f1, 99);

    printf("Nova fila: ");
    imprimir(f1);
    printf("\nNova quantidade de elementos: %d\n", quantidade(f1));

    f1 = libera(f1);
    printf("\n");
    return 0;
}

Fila* cria() {
    Fila* f = (Fila*)malloc(sizeof(Fila));
```

```
    if (f == NULL) {
        exit(1);
    }
    f->ini = f->fim = f->cont = 0;
    return f;
}

int incr(int i) {
    if (i == N - 1)
        return 0;
    else
        return i + 1;
}

int decr(int i) {
    if (i == 0)
        return N - 1;
    else
        return i - 1;
}

void inserir(Fila* f, int v) {
    if (f->cont < N) {
        f->vet[f->fim] = v;
        f->fim = incr(f->fim);
        f->cont++;
    } else {
        printf("\nCapacidade maxima da fila atingida!\n");
    }
}

void retirar(Fila* f) {
    if (f->cont != 0) {
        f->ini = incr(f->ini);
        f->cont--;
    } else {
        printf("Fila vazia. Impossivel retirar.");
    }
}

void imprimir(Fila* f) {
    if (f->cont == 0) {
        printf("Fila esta vazia.");
        return;
    }
    int i;
    int pos = f->ini;
    for (i = 0; i < f->cont; i++) {
        printf("[%d] ", f->vet[pos]);
    }
}
```

```
        pos = incr(pos);
    }
}

Fila* libera(Fila* f) {
    free(f);
    return NULL;
}

int quantidade(Fila* f) {
    return f->cont;
}

void fura_fila(Fila* f, int x) {
    if (f->cont >= N) {
        printf("Fila cheia! Impossivel furar a fila.\n");
        return;
    }
    f->ini = decr(f->ini);
    f->vet[f->ini] = x;
    f->cont++;
}
```