

main.c - FILA COM UNION

```
1 //programa de fila
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <conio.h>
6 #include "fila.h"
7
8 void ler(elem *e, int tipo) {
9     if (tipo == 0)
10        scanf("%d", &e->i);
11    else
12        scanf("%f", &e->r);
13 }
14
15 void escrever(elem *e, int tipo) {
16     if (tipo == 0)
17        printf("(%d)", e->i);
18    else
19        printf("(%f)", e->r);
20 }
21
22 int main(void) {
23     elem ch, x;
24     int erro, tipo;
25     Fila F;
26
27     printf("Com que tipo de elemento quer trabalhar
28 (0=inteiro|1=real)? ");
29     scanf("%d", &tipo);
30     if (tipo == 0)
31        Create(&F, inteiro);
32     else
33        Create(&F, real);
34
35     printf("Entre com um elemento: ");
36     ler(&ch, tipo);
37     Entra(&F, &ch, &erro);
38     if (erro)
39        printf("(erro)\n");
40     printf("Entre com outro elemento: ");
41     ler(&ch, tipo);
42     Entra(&F, &ch, &erro);
43     if (erro)
44        printf("(erro)\n");
45     printf("Entre com mais um elemento: ");
46     ler(&ch, tipo);
47     Entra(&F, &ch, &erro);
48     if (erro)
49        printf("(erro)\n");
50     printf("Tirando um elemento: ");
51     Sai(&F, &x, &erro);
52     if (erro)
53        printf("(erro)\n");
54     else
55        escrever(&x, tipo);
56
57     printf("\nTirando o resto da fila: ");
58     while (!IsEmpty(F)) {
59         Sai(&F, &x, &erro);
```

```
59         if (erro)
60             printf("(erro) ");
61         else
62             escrever(&x, tipo);
63     }
64
65     return 0;
66 }
67
```

fila.h

```
1  #define TamFila 100
2
3  typedef union {
4      int i;
5      float r;
6  } elem;
7
8  typedef struct {
9      int inicio, fim, total;
10     elem itens[TamFila];
11     enum tipo {
12         inteiro, real
13     } Tipo;
14 } Fila;
15
16 void Create(Fila*, int);
17 void Empty(Fila*);
18 int IsEmpty(Fila);
19 int IsFull(Fila);
20 void Entra(Fila*, elem*, int*);
21 void Sai(Fila*, elem*, int*);
22
```

fila.c

```
1  #include <string.h>
2  #include "fila.h"
3
4  void Create(Fila *F, int tipo) {
5      F->inicio = 0;
6      F->fim = 0;
7      F->total = 0;
8      F->Tipo = tipo;
9  }
10
11 void Empty(Fila *F) {
12     F->inicio = 0;
13     F->fim = 0;
14     F->total = 0;
15 }
16
17 int IsEmpty(Fila F) {
18     if (F.total == 0)
19         return 1;
20     else
21         return 0;
22 }
23
24 int IsFull(Fila F) {
25     if (F.total == TamFila)
26         return 1;
27     else
28         return 0;
29 }
30
31 void Entra(Fila *F, elem *X, int *erro) {
32     if (!IsFull(*F)) {
33         *erro = 0;
34         F->total++;
35         if (F->Tipo == inteiro)
36             F->itens[F->fim].i = X->i;
37         else
38             F->itens[F->fim].r = X->r;
39         if (F->fim == TamFila - 1)
40             F->fim = 0;
41         else
42             F->fim++;
43     }
44     else
45         *erro = 1;
46     return;
47 }
48
49 void Sai(Fila *F, elem *X, int *erro) {
50     if (!IsEmpty(*F)) {
51         *erro = 0;
52         F->total--;
53         if (F->Tipo == inteiro)
54             X->i = F->itens[F->inicio].i;
55         else
56             X->r = F->itens[F->inicio].r;
57         if (F->inicio == TamFila - 1)
58             F->inicio = 0;
59         else
```

```
60         F->inicio++;
61     }
62     else
63         *erro = 1;
64     return;
65 }
66
```