

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct filaNo {
5      int data;
6      struct filaNo *proximoPonteiro;
7  };
8
9  typedef struct filaNo FilaNo;
10 typedef FilaNo *FilaNoPonteiro;
11
12 void imprimirFila (FilaNoPonteiro atualPonteiro);
13 int estaVazia (FilaNoPonteiro cabecaPonteiro);
14 int desenfileirar (FilaNoPonteiro *cabecaPonteiro, FilaNoPonteiro *caudaPonteiro);
15 void enfileirar (FilaNoPonteiro *cabecaPonteira, FilaNoPonteiro *caudaPonteiro, int value);
16 void instrucoes (void);
17
18 int main()
19 {
20     FilaNoPonteiro cabecaPonteiro = NULL;
21     FilaNoPonteiro caudaPonteiro = NULL;
22     int escolha;
23     int valor;
24
25     instrucoes ();
26     printf("?\\n");
27     scanf ("%d", &escolha);
28
29     //enquanto o usuário não digitar 4
30     while (escolha != 4) {
31         switch (escolha) {
32             case 1: //enfileira valor
33                 printf ("Digite um valor: ");
34                 scanf ("%d", &valor);
35                 enfileirar (&cabecaPonteiro, &caudaPonteiro, valor);
36                 break;
37             case 2: //desenfileira valor
38                 //se a fila não estiver vazia
39                 if (!estaVazia(cabecaPonteiro)) {
40                     printf ("%d saiu da fila!\\n", desenfileirar(&cabecaPonteiro, &caudaPonteiro));
41                 }
42                 break;
43             case 3: //imprime a fila
44                 imprimirFila (cabecaPonteiro);
45                 break;
46             default:
47                 printf("Escolha invalida!\\n\\n");
48                 instrucoes();
49                 break;
50         }
51         printf("?\\n");
52         scanf ("%d", &escolha);
53     }
54     printf("Fim da execucao!\\n");
55     return 0;
56 }
57
58
59 void instrucoes (void) {
60     printf("Digite sua escolha:\\n"
61         "    1 para incluir um item na fila.\\n"
62         "    2 para remover um item da fila.\\n"
63         "    3 para imprimir a fila.\\n"
64         "    4 para encerrar.\\n");
65 }
66

```

```

67 //insere um nó na cauda da fila
68 void enfileirar (FilaNoPonteiro *cabecaPonteiro, FilaNoPonteiro *caudaPonteiro, int value) {
69     FilaNoPonteiro novoPonteiro; //ponteiro para um novo nó
70     novoPonteiro = malloc (sizeof (FilaNo));
71
72     if (novoPonteiro != NULL) {
73         novoPonteiro->data = value;
74         novoPonteiro->proximoPonteiro = NULL;
75
76         //se estiver vazia insere nó na cabeça
77         if (estaVazia (*cabecaPonteiro)) {
78             *cabecaPonteiro = novoPonteiro;
79         }
80         else {
81             (*caudaPonteiro)->proximoPonteiro = novoPonteiro;
82         }
83         *caudaPonteiro = novoPonteiro;
84     }
85     else {
86         printf ("%d nao inserido. Nao ha memoria disponivel.\n", value);
87     }
88 }
89
90 //remove nó da cabeça da fila
91 int desenfileirar (FilaNoPonteiro *cabecaPonteiro, FilaNoPonteiro *caudaPonteiro) {
92     int valor; //valor do nó
93     FilaNoPonteiro tempPonteiro; //ponteiro de nó temporário
94
95     valor = (*cabecaPonteiro) -> data;
96     tempPonteiro = *cabecaPonteiro;
97     *cabecaPonteiro = (*cabecaPonteiro) -> proximoPonteiro;
98
99     //se a fila estiver vazia
100    if (*cabecaPonteiro == NULL) {
101        *caudaPonteiro = NULL;
102    }
103
104    free (tempPonteiro);
105    return valor;
106 }
107
108 //retorna 1 se a lista estiver vazia; caso contrário, retorna 0
109 int estaVazia (FilaNoPonteiro cabecaPonteiro) {
110     return cabecaPonteiro == NULL;
111 }
112
113 //imprime a fila
114 void imprimirFila (FilaNoPonteiro atualPonteiro) {
115     //se a fila estiver vazia
116     if (atualPonteiro == NULL) {
117         printf ("A fila esta vazia!\n\n");
118     }
119     else {
120         printf("A fila e:\n");
121         //enquanto não for fim da fila
122         while (atualPonteiro != NULL) {
123             printf("%d --> ", atualPonteiro -> data);
124             atualPonteiro = atualPonteiro -> proximoPonteiro;
125         }
126
127         printf ("NULL\n\n");
128     }
129 }

```