



Universidade Federal de São João del Rei
Departamento de Ciência da Computação
Curso de Ciência da Computação

Roteiro 5

Adélson de Oliveira Carmo Júnior
212050019

1 Fila

1.1 Fila Sequencial Estática

1.1.1 Código

```
1 #ifndef FILA_H
2 #define FILA_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 #define MAX 100
8 typedef struct{
9     int qtd, ini, fim;
10    int dados[MAX];
11 }Fila;
12
13
14 Fila* criaFila();
15 void destroiFila(Fila *fi);
16 int tamanhoFila(Fila *fi);
17 int estaCheia(Fila *fi);
18 int estaVazia(Fila *fi);
19 int enfileirar(Fila* fi, int elem);
20 int desenfileirar(Fila* fi);
21 int verInicio(Fila* fi, int* p);
22 void imprime(Fila* fi);
23
24 /*Funcao extra*/
25 void aguardaLimpa();
26
27 #endif
```

codigos/questao11/questao_11.h

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_11.h"
4
5
6 Fila* criaFila(){
7     Fila* fi;
8     fi = (Fila*) malloc (sizeof(Fila));
9     if(fi != NULL){
10         fi->qtd = fi->ini = fi->fim = 0;
11     }
12     return fi;
13 }
14
15 void destroiFila(Fila *fi){
16     if(fi != NULL)
17         free(fi);
18 }
19
20 int tamanhoFila(Fila *fi){
21     if(fi == NULL)
22         return -1;
23     return fi->qtd;
24 }
25
26 int estaCheia(Fila *fi){
27     if(fi == NULL)
28         return -1;
29     return (fi->qtd == MAX);
30 }
31
32 int estaVazia(Fila *fi){
33     if(fi == NULL)
34         return -1;
35     return (fi->qtd == 0);
36 }
```

```

37
38
39 int enfileirar(Fila* fi, int elem){
40     if(fi == NULL) return 0;
41     if(estaCheia(fi)) return 0;
42     fi->dados[fi->fim] = elem;
43     fi->fim = (fi->fim+1) % MAX;
44     fi->qtd++;
45     return 1;
46 }
47
48 int desenfileirar(Fila* fi){
49     if(fi == NULL) return 0;
50     if(estaVazia(fi)) return 0;
51     fi->ini = (fi->ini+1) % MAX;
52     fi->qtd--;
53     return 1;
54 }
55
56 int verInicio(Fila* fi, int* p){
57     if(fi == NULL) return 0;
58     if(estaVazia(fi)) return 0;
59     *p = fi->dados[fi->ini];
60     return 1;
61 }
62

```

```

63 void imprime(Fila* fi){
64     if(fi == NULL) return;
65     if(estaVazia(fi)){
66         printf("Fila Vazia!\n");
67         return;
68     }
69     int i = fi->ini;
70     printf("Elementos: \n");
71     do{
72         printf("%d ", fi->dados[i]);
73         i = (i + 1) % MAX;
74     }while(i != fi->fim);
75     printf("\n");
76 }
77
78 /* Funcao para esperar resposta do
   usuario e depois limpar*/
79 void aguardaLimpa(){
80     getchar();
81     printf("\n\nAperte qualquer tecla
   para continuar\n");
82     getchar();
83     system("clear");
84 }

```

codigos/questao11/questao-11.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_11.h"
4
5 int main(){
6     int escolha_inicial, item;
7     int *inicio =
8         (int*)malloc(sizeof(int));
9     Fila *fila;
10     do{
11         printf("0 que deseja fazer?\n1
   - Criar fila\n2 - Enfileirar
   um item\n3 - Ver o inicio da
   fila\n"
12         "4 - Desenfileirar um item;\n5
   - Imprimir a fila;\n6 -
   Destruir a fila\n7 -
   Sair\n");
13         scanf("%d",&escolha_inicial);
14         switch (escolha_inicial){
15             case 7:
16                 break;
17             case 1:
18                 fila = criaFila();
19                 printf("Fila criada!");
20                 aguardaLimpa();
21                 break;
22

```

```

23
24         case 2:
25             printf("Digite o numero que
   deseja enfileirar");
26             scanf("%d",&item);
27             if(enfileirar(fila,item) ==
   1)
28                 printf("Elemento
   enfileirado!");
29             else
30                 printf("Fila cheia ou
   nao existente!");
31             aguardaLimpa();
32             break;
33         case 3:
34             if(verInicio(fila, inicio)
   == 1)
35                 printf("0 valor inicial
   da fila eh: %d",
   *inicio);
36             else
37                 printf("Fila Vazia");
38             aguardaLimpa();
39             break;
40         case 4:
41             if(desenfileirar(fila) == 1)
42                 printf("Primeiro

```

```

45         elemento da fila
46         desenfilerado!");
47     else
48         printf("Fila Vazia!");
49         aguardaLimpa();
50         break;
51     case 5:
52         imprime(fila);
53         aguardaLimpa();
54         break;
55     case 6:
56         destroiFila(fila);
57         printf("Fila destruida");
58         aguardaLimpa();
59         break;
60     default:
61         printf("Valor invalido");
62         break;
63     }
64 }while(escolha_inicial != 7);
65 free(inicio);
66 return 0;
67 }
68
69
70
codigos/questao11/main.c

```

```

1 all: questao_11.o
2 gcc questao_11.o main.c -o main
3
4 questao_11.o: questao_11.h questao_11.c
5 gcc -c questao_11.c
6
7 clean:
8 rm -f questao_11.o main
codigos/questao11/Makefile

```

1.1.2 Saída

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
1
Fila criada!
Aperte qualquer tecla para continuar

```

Figura 1: Questão 1.1 - Cria Fila

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
2
Digite o numero que deseja enfileirar5
Elemento enfileirado!
Aperte qualquer tecla para continuar

```

Figura 2: Questão 1.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
2
Digite o numero que deseja enfileirar?
Elemento enfileirado!

Aperte qualquer tecla para continuar

```

Figura 3: Questão 1.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
3
0 valor inicial da fila eh: 5

Aperte qualquer tecla para continuar

```

Figura 4: Questão 1.1 - Ver Inicio

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
5
Elementos:
5 7

Aperte qualquer tecla para continuar

```

Figura 5: Questão 1.1 - Imprime

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
4
Primeiro elemento da fila desenfileirado!

Aperte qualquer tecla para continuar

```

Figura 6: Questão 1.1 - Desenfilera

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
5
Elementos:
7

Aperte qualquer tecla para continuar

```

Figura 7: Questão 1.1 - Imprime

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
6
Fila destruida

Aperte qualquer tecla para continuar

```

Figura 8: Questão 1.1 - Destrói Fila

1.2 Fila Simplesmente Encadeada

Código

```

1 #ifndef FSE_H
2 #define FSE_H
3
4 #include <stdio.h>

```

```

5 #include <stdlib.h>
6
7 typedef struct NO {
8     int info ;

```

```

9     struct NO *prox ;
10 } NO ;
11
12 typedef struct {
13     int qtd ;
14     struct NO *ini ;
15     struct NO *fim ;
16 } Fila ;
17
18
19 Fila* criaFila();
20 void destroiFila(Fila *);
21 int tamanhoFila(Fila *);

```

```

22 int estaCheia(Fila *);
23 int estaVazia(Fila *);
24 int enfileirar(Fila *, int);
25 int desenfileirar(Fila *);
26 int verInicio(Fila *, int*);
27 void imprime(Fila *);
28
29 /*Funcao extra*/
30 void aguardaLimpa();
31
32 # endif

```

codigos/questao12/questao_12.h

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_12.h"
4
5 Fila* criaFila(){
6     Fila* fi;
7     fi = (Fila*) malloc (sizeof(Fila));
8     if(fi != NULL){
9         fi->qtd = 0;
10        fi->ini = fi->fim = NULL;
11    }
12    return fi;
13 }
14
15 void destroiFila(Fila *fi){
16     if(fi != NULL) {
17         while (fi->ini != NULL) {
18             NO* aux = fi->ini;
19             fi->ini = fi->ini->prox;
20             free(aux);
21         }
22         free(fi);
23     }
24 }
25
26 int tamanhoFila(Fila *fi){
27     if(fi == NULL)
28         return -1;
29     return fi->qtd;
30 }
31
32 int estaCheia(Fila *fi){
33     return 0; // Em uma fila encadeada,
                // ela nunca fica cheia
34 }
35
36 int estaVazia(Fila *fi){
37     if(fi == NULL)
38         return -1;
39     return (fi->qtd == 0);
40 }
41

```

```

42 int enfileirar(Fila* fi, int elem){
43     if(fi == NULL) return 0;
44
45     NO* novo = (NO*) malloc(sizeof(NO));
46     if(novo == NULL) return 0;
47
48     novo->info = elem;
49     novo->prox = NULL;
50
51     if(estaVazia(fi)){
52         fi->ini = fi->fim = novo;
53     } else {
54         fi->fim->prox = novo;
55         fi->fim = novo;
56     }
57
58     fi->qtd++;
59     return 1;
60 }
61
62 int desenfileirar(Fila* fi){
63     if(fi == NULL || estaVazia(fi))
64         return 0;
65
66     NO* aux = fi->ini;
67     fi->ini = fi->ini->prox;
68
69     if(fi->ini == NULL)
70         fi->fim = NULL;
71
72     free(aux);
73     fi->qtd--;
74     return 1;
75 }
76
77 int verInicio(Fila* fi, int* p){
78     if(fi == NULL || estaVazia(fi))
79         return 0;
80     *p = fi->ini->info;
81     return 1;
82 }
83

```

```

82 void imprime(Fila* fi){
83     if(fi == NULL) return;
84     if(estaVazia(fi)){
85         printf("Fila Vazia!\n");
86         return;
87     }
88     NO* aux = fi->ini;
89     printf("Elementos: \n");
90     while(aux != NULL){
91         printf("%d ", aux->info);
92         aux = aux->prox;
93     }
94     printf("\n");

```

```

95 }
96
97 /* Funcao para esperar resposta do
   usuario e depois limpar*/
98 void aguardaLimpa(){
99     getchar();
100    printf("\n\nAperte qualquer tecla
       para continuar\n");
101    getchar();
102    system("clear");
103 }

```

codigos/questao12/questao_12.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "questao_12.h"
4
5  int main(){
6      int escolha_inicial, item;
7      int *inicio =
        (int*)malloc(sizeof(int));
8      Fila *fila;
9      do{
10         printf("0 que deseja fazer?\n1
           - Criar fila\n2 - Enfileirar
           um item\n3 - Ver o inicio da
           fila\n"
11         "4 - Desenfileirar um item;\n5
           - Imprimir a fila;\n6 -
           Destruir a fila\n7 -
           Sair\n");
12         scanf("%d",&escolha_inicial);
13         switch (escolha_inicial){
14
15             case 7:
16                 break;
17
18             case 1:
19                 fila = criaFila();
20                 printf("Fila criada!");
21                 aguardaLimpa();
22                 break;
23
24             case 2:
25                 printf("Digite o numero que
                   deseja enfileirar");
26                 scanf("%d",&item);
27                 if(enfileirar(fila,item) ==
                   1)
28                     printf("Elemento
                       enfileirado!");
29                 else
30                     printf("Fila cheia ou
                       nao existente!");
31                 aguardaLimpa();

```

```

32         break;
33
34         case 3:
35             if(verInicio(fila, inicio)
               == 1)
36                 printf("0 valor inicial
                   da fila eh: %d",
                   *inicio);
37             else
38                 printf("Fila Vazia");
39             aguardaLimpa();
40             break;
41
42         case 4:
43             if(desenfileirar(fila) == 1)
44                 printf("Primeiro
                   elemento da fila
                   desenfileirado!");
45             else
46                 printf("Fila Vazia!");
47             aguardaLimpa();
48             break;
49
50         case 5:
51             imprime(fila);
52             aguardaLimpa();
53             break;
54
55         case 6:
56             destroiFila(fila);
57             printf("Fila destruida");
58             aguardaLimpa();
59             break;
60
61         default:
62             printf("Valor invalido");
63             break;
64     }
65 }while(escolha_inicial != 7);
66 free(inicio);
67
68

```

```
69     return 0;
```

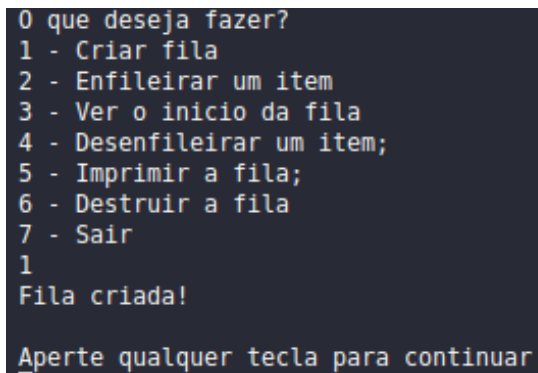
```
70 }
```

codigos/questao12/main.c

```
1 all: questao_12.o
2   gcc questao_12.o main.c -o main
3
4 questao_12.o: questao_12.h questao_12.c
5   gcc -c questao_12.c
```

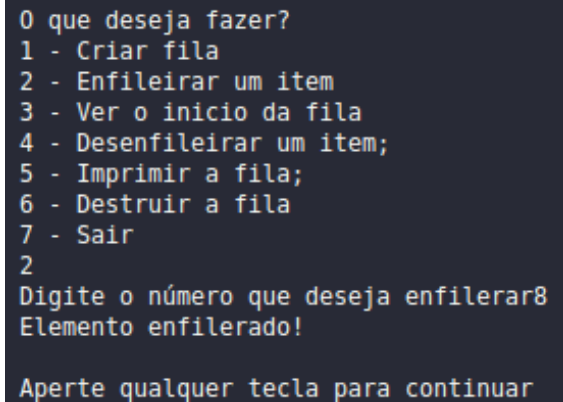
```
6
7 clean:
8   rm -f questao_12.o main
codigos/questao12/Makefile
```

1.2.1 Saída



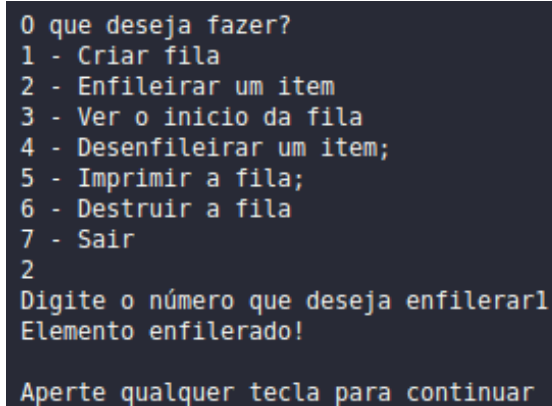
```
0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
1
Fila criada!
Aperte qualquer tecla para continuar
```

Figura 9: Questão 1.2 - Cria Fila



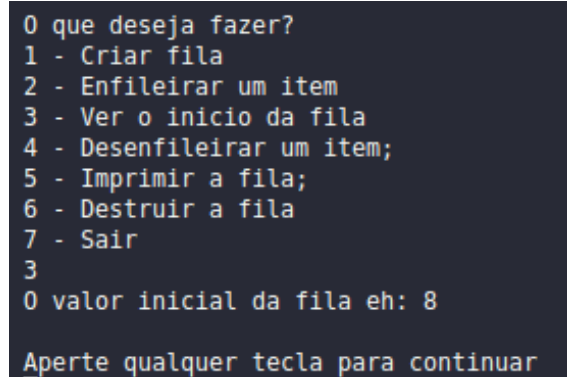
```
0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
2
Digite o número que deseja enfileirar: 8
Elemento enfileirado!
Aperte qualquer tecla para continuar
```

Figura 10: Questão 1.2 - Adiciona Elemento



```
0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
2
Digite o número que deseja enfileirar: 1
Elemento enfileirado!
Aperte qualquer tecla para continuar
```

Figura 11: Questão 1.2 - Adiciona Elemento



```
0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
3
0 valor inicial da fila eh: 8
Aperte qualquer tecla para continuar
```

Figura 12: Questão 1.2 - Ver Inicio

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
5
Elementos:
8 1

Aperte qualquer tecla para continuar

```

Figura 13: Questão 1.2 - Imprime

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
4
Primeiro elemento da fila desenfileirado

Aperte qualquer tecla para continuar

```

Figura 14: Questão 1.2 - Desenfilera

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
5
Elementos:
1

Aperte qualquer tecla para continuar

```

Figura 15: Questão 1.2 - Imprime

```

0 que deseja fazer?
1 - Criar fila
2 - Enfileirar um item
3 - Ver o inicio da fila
4 - Desenfileirar um item;
5 - Imprimir a fila;
6 - Destruir a fila
7 - Sair
6
Fila destruida

Aperte qualquer tecla para continuar

```

Figura 16: Questão 1.2 - Destrói Fila

2 Pilha

2.1 Pilha Sequencial Estática

2.1.1 Código

```

1 #ifndef PILHA_H
2 #define PILHA_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 #define MAX 100
8 typedef struct{
9     int topo;
10    int dados[MAX];
11 }Pilha;
12
13
14 Pilha* criaPilha();
15 void destroiPilha(Pilha *);

```

```

16 int tamanhoPilha(Pilha *);
17 int estaCheia(Pilha *);
18 int estaVazia(Pilha *);
19 int empilhar(Pilha *, int);
20 int desempilhar(Pilha *);
21 int verTopo(Pilha *, int *);
22 void imprime(Pilha *);
23
24 /* Funcao extra */
25 void aguardaLimpa();
26
27 #endif

```

codigos/questao21/questao_21.h


```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_21.h"
4
5 Pilha* criaPilha(){
6     Pilha* pi;
7     pi = (Pilha*) malloc
8         (sizeof(Pilha));
9     if(pi != NULL){
10         pi->topo = 0;
11     }
12     return pi;
13 }
14 void destroiPilha(Pilha *pi){
15     if(pi != NULL)
16         free(pi);
17 }
18
19 int tamanhoPilha(Pilha *pi){
20     if(pi == NULL)
21         return -1;
22     return pi->topo;
23 }
24
25 int estaCheia(Pilha *pi){
26     if(pi == NULL)
27         return -1;
28     return (pi->topo == MAX);
29 }
30
31 int estaVazia(Pilha *pi){
32     if(pi == NULL)
33         return -1;
34     return (pi->topo == 0);
35 }
36
37
38 int empilhar(Pilha* pi, int elem){
39     if(pi == NULL) return 0;
40     if(estaCheia(pi)) return 0;
41     pi->dados[pi->topo] = elem;

```

```

42     pi->topo++;
43     return 1;
44 }
45
46 int desempilhar(Pilha* pi){
47     if(pi == NULL) return 0;
48     if(estaVazia(pi)) return 0;
49     pi->topo--;
50     return 1;
51 }
52
53 int verTopo(Pilha* pi, int* p){
54     if(pi == NULL) return 0;
55     if(estaVazia(pi)) return 0;
56     *p = pi->dados[pi->topo-1];
57     return 1;
58 }
59
60 void imprime(Pilha* pi){
61     if(pi == NULL) return;
62     if(estaVazia(pi)){
63         printf("Pilha Vazia!\n");
64         return;
65     }
66     printf("Elementos: \n");
67     int i;
68     for(i=pi->topo-1; i>=0; i--){
69         printf("%d ", pi->dados[i]);
70         printf("\n");
71     }
72
73     /* Funcao para esperar resposta do
74        usuario e depois limpar*/
74 void aguardaLimpa(){
75     getchar();
76     printf("\n\nAperte qualquer tecla
77         para continuar\n");
77     getchar();
78     system("clear");
79 }

```

codigos/questao21/questao_21.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_21.h"
4
5 int main(){
6     int escolha_inicial, item;
7     int *topo =
8         (int*)malloc(sizeof(int));
9     Pilha *pilha;
10
11     do{
12         printf("0 que deseja fazer?\n1
13             - Criar pilha\n2 - Empilhar

```

```

12         um item\n3 - Ver o topo da
13         pilha\n"
14         "4 - Desempilha um item;\n5 -
15         Imprimir a pilha;\n6 -
16         Destruir a pilha\n7 -
17         Sair\n");
18     scanf("%d",&escolha_inicial);
19     switch (escolha_inicial){
20
21         case 7:
22             break;
23
24         case 1:

```

```

20     pilha = criaPilha();          45         printf("Ultimo elemento
21     printf("Pilha criada!");      da pilha
22     aguardaLimpa();              desempilhado!");
23     break;                        46     else
24                                     47         printf("Pilha Vazia!");
25     case 2:                        48     aguardaLimpa();
26     printf("Digite o numero que  49     break;
27         deseja enfileirar");      50
28     scanf("%d",&item);            51     case 5:
29     if(empilhar(pilha,item) ==    52         imprime(pilha);
30         1)                          53         aguardaLimpa();
31         printf("Elemento          54         break;
32         enfileirado!");           55
33     else                           56     case 6:
34         printf("Pilha cheia ou    57         destroiPilha(pilha);
35         nao existente!");         58         printf("Pilha destruida");
36     aguardaLimpa();               59         aguardaLimpa();
37     break;                        60         break;
38                                     61
39     case 3:                        62     default:
40     if(verTopo(pilha, topo) ==    63         printf("Valor invalido");
41         1)                          64         break;
42         printf("O valor do topo   65     }
43         da pilha eh: %d",         66     }while(escolha_inicial != 7);
44         *topo);                   67
45     else                           68     free(topo);
46         printf("Pilha Vazia");    69
47     aguardaLimpa();               70     return 0;
48     break;                        71 }
49
50                                     codigos/questao21/main.c
51     case 4:
52     if(desempilhar(pilha) == 1)

```

```

1 all: questao_21.o                6
2 gcc questao_21.o main.c -o main  7 clean:
3                                     8 rm -f questao_21.o main
4 questao_21.o: questao_21.h questao_21.c
5 gcc -c questao_21.c              codigos/questao21/Makefile

```

2.1.2 Saída

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
1
Pilha criada!

Aperte qualquer tecla para continuar

```

Figura 17: Questão 2.1 - Cria Pilha

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
2
Digite o numero que deseja enfileirar9
Elemento enfileirado!

Aperte qualquer tecla para continuar

```

Figura 18: Questão 2.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
2
Digite o numero que deseja enfileirar6
Elemento enfileirado!

Aperte qualquer tecla para continuar

```

Figura 19: Questão 2.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
3
0 valor do topo da pilha eh: 6

Aperte qualquer tecla para continuar

```

Figura 20: Questão 2.1 - Ver Topo

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
5
Elementos:
6 9

Aperte qualquer tecla para continuar

```

Figura 21: Questão 2.1 - Imprime

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
4
Ultimo elemento da pilha desempilhado!

Aperte qualquer tecla para continuar

```

Figura 22: Questão 2.1 - Desempilha

2.2 Pilha Simplesmente Encadeada

2.2.1 Código

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
5
Elementos:
9

Aperte qualquer tecla para continuar

```

Figura 23: Questão 2.1 - Imprime

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
6
Pilha destruida

Aperte qualquer tecla para continuar

```

Figura 24: Questão 2.1 - Destrói Pilha

```

1 #ifndef PSE_H
2 #define PSE_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 typedef struct NO {
8     int info ;
9     struct NO *prox ;
10 } NO ;
11
12 typedef struct {
13     int qtd ;
14     struct NO *topo ;
15 } Pilha ;
16
17

```

```

18 Pilha* criaPilha();
19 void destroiPilha(Pilha *);
20 int tamanhoPilha(Pilha *);
21 int estaCheia(Pilha *);
22 int estaVazia(Pilha *);
23 int empilhar(Pilha *, int);
24 int desempilhar(Pilha *);
25 int verTopo(Pilha *, int *);
26 void imprime(Pilha *);
27
28 /* Funcao extra */
29 void aguardaLimpa();
30
31 #endif

```

codigos/questao22/questao_22.h

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_22.h"
4
5 Pilha* criaPilha(){
6     Pilha* pi;
7     pi = (Pilha*) malloc
8         (sizeof(Pilha));
9     if(pi != NULL){
10         pi->qtd = 0;
11         pi->topo = NULL;
12     }
13     return pi;
14 }
15 void destroiPilha(Pilha *pi){
16     if(pi != NULL) {
17         while (pi->topo != NULL) {
18             NO* aux = pi->topo;
19             pi->topo = pi->topo->prox;

```

```

20             free(aux);
21         }
22         free(pi);
23     }
24 }
25
26 int tamanhoPilha(Pilha *pi){
27     if(pi == NULL)
28         return -1;
29     return pi->qtd;
30 }
31
32 int estaCheia(Pilha *pi){
33     return 0;
34 }
35
36 int estaVazia(Pilha *pi){
37     if(pi == NULL)
38         return -1;
39     return (pi->qtd == 0);

```

```

40 }
41
42 int empilhar(Pilha* pi, int elem){
43     if(pi == NULL) return 0;
44
45     NO* novo = (NO*) malloc(sizeof(NO));
46     if(novo == NULL) return 0;
47
48     novo->info = elem;
49     novo->prox = pi->topo;
50
51     pi->topo = novo;
52     pi->qtd++;
53
54     return 1;
55 }
56
57 int desempilhar(Pilha* pi){
58     if(pi == NULL || estaVazia(pi))
59         return 0;
60
61     NO* aux = pi->topo;
62     pi->topo = pi->topo->prox;
63     free(aux);
64     pi->qtd--;
65
66     return 1;
67 }
68
69 int verTopo(Pilha* pi, int* p){
70     if(pi == NULL || estaVazia(pi))
71         return 0;

```

```

70     *p = pi->topo->info;
71     return 1;
72 }
73
74 void imprime(Pilha* pi){
75     if(pi == NULL) return;
76     if(estaVazia(pi)){
77         printf("Pilha Vazia!\n");
78         return;
79     }
80
81     NO* aux = pi->topo;
82     printf("Elementos: \n");
83     while(aux != NULL){
84         printf("%d ", aux->info);
85         aux = aux->prox;
86     }
87     printf("\n");
88 }
89
90 /* Funcao para esperar resposta do
   usuario e depois limpar*/
91 void aguardaLimpa(){
92     getchar();
93     printf("\n\nAperte qualquer tecla
   para continuar\n");
94     getchar();
95     system("clear");
96 }

```

codigos/questao22/questao_22.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao_22.h"
4
5 int main(){
6     int escolha_inicial, item;
7     int *topo =
8         (int*) malloc(sizeof(int));
9     Pilha *pilha;
10
11     do{
12         printf("0 que deseja fazer?\n1
   - Criar pilha\n2 - Empilha
   um item\n3 - Ver o topo da
   pilha\n"
13         "4 - Desempilha um item;\n5 -
   Imprimir a pilha;\n6 -
   Destruir a pilha\n7 -
   Sair\n");
14         scanf("%d",&escolha_inicial);
15         switch (escolha_inicial){
16             case 7:
17                 break;

```

```

18
19         case 1:
20             pilha = criaPilha();
21             printf("Pilha criada!");
22             aguardaLimpa();
23             break;
24
25         case 2:
26             printf("Digite o numero que
   deseja enfileirar");
27             scanf("%d",&item);
28             if(empilhar(pilha,item) ==
   1)
29                 printf("Elemento
   enfileirado!");
30
31             else
32                 printf("Pilha cheia ou
   nao existente!");
33             aguardaLimpa();
34             break;
35
36         case 3:
37             if(verTopo(pilha, topo) ==
   1)

```

```

37         printf("0 valor do topo da pilha eh: %d", *topo);
38     else
39         printf("Pilha Vazia");
40     aguardaLimpa();
41     break;
42
43     case 4:
44         if(desempilhar(pilha) == 1)
45             printf("Ultimo elemento da pilha
desempilhado!");
46     else
47         printf("Pilha Vazia!");
48     aguardaLimpa();
49     break;
50
51     case 5:
52         imprime(pilha);
53         aguardaLimpa();
54         break;
55
56     case 6:
57         destroiPilha(pilha);
58         printf("Pilha destruida");
59         aguardaLimpa();
60         break;
61
62     default:
63         printf("Valor invalido");
64         break;
65     }
66 }while(escolha_inicial != 7);
67 free(topo);
68
69 return 0;
70
71 }

```

codigos/questao22/main.c

```

1 all: questao_22.o
2 gcc questao_22.o main.c -o main
3
4 questao_22.o: questao_22.h questao_22.c
5 gcc -c questao_22.c
6
7 clean:
8 rm -f questao_22.o main

```

codigos/questao22/Makefile

2.2.2 Saída

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
1
Pilha criada!

Aperte qualquer tecla para continuar

```

Figura 25: Questão 2.2 - Cria Pilha

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
2
Digite o numero que deseja enfileirar2
Elemento enfileirado!

Aperte qualquer tecla para continuar

```

Figura 26: Questão 2.2 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
2
Digite o numero que deseja enfileirar3
Elemento enfileirado!

Aperte qualquer tecla para continuar

```

Figura 27: Questão 2.2 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
3
0 valor do topo da pilha eh: 3

Aperte qualquer tecla para continuar

```

Figura 28: Questão 2.2 - Ver Topo

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
5
Elementos:
3 2

Aperte qualquer tecla para continuar

```

Figura 29: Questão 2.2 - Imprime

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
4
Ultimo elemento da pilha desempilhado!

Aperte qualquer tecla para continuar

```

Figura 30: Questão 2.2 - Desempilha

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
5
Elementos:
2

Aperte qualquer tecla para continuar

```

Figura 31: Questão 2.2 - Imprime

```

0 que deseja fazer?
1 - Criar pilha
2 - Empilha um item
3 - Ver o topo da pilha
4 - Desempilha um item;
5 - Imprimir a pilha;
6 - Destruir a pilha
7 - Sair
6
Pilha destruida

Aperte qualquer tecla para continuar

```

Figura 32: Questão 2.2 - Destrói Pilha