



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

Roteiro 6

Adélson de Oliveira Carmo Júnior
212050019

1 Deque

1.1 Deque Sequencial Estático

Código

```
1 #ifndef DEQUE_H
2 #define DEQUE_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 }Deque;
12
13 /* Funcoes */
14 Deque* criaDeque();
15 void destroiDeque(Deque *);
16 int tamanhoDeque(Deque *);
17 int estaCheio(Deque *);
18 int estaVazio(Deque *);
19 int insereIni(Deque *, int);
20 int insereFim(Deque *, int);
21 int removeIni(Deque *);
22 int removeFim(Deque *);
23 int verInicio(Deque *, int *);
24 int verFim(Deque *, int *);
25 void imprime(Deque *);
26
27 /*Funcao extra*/
28 void aguardaLimpa();
29
30 #endif
31
32 codigos/questao11/questao11.h
33
34 1 #include <stdio.h>
35 2 #include <stdlib.h>
36 3 #include "questao11.h"
37 4
38 5 Deque* criaDeque(){
39 6     Deque* dq;
40 7     dq = (Deque*) malloc
41         (sizeof(Deque));
42 8     if(dq != NULL){
43 9         dq->qtd = dq->ini = dq->fim = 0;
44 10    }
45 11    return dq;
46 12 }
47 13
48 14 void destroiDeque(Deque *dq){
49 15     if(dq != NULL)
50 16         free(dq);
51 17 }
52 18
53 19 int tamanhoDeque(Deque *dq){
54 20     if(dq == NULL)
55 21         return -1;
56 22     return dq->qtd;
57 23 }
58 24
59 25 int estaCheio(Deque *dq){
60 26     if(dq == NULL)
61 27         return -1;
62 28     return (dq->qtd == MAX);
63 29 }
64 30
65 31 int estaVazio(Deque *dq){
```

```

32     if(dq == NULL)
33         return -1;
34     return (dq->qtd == 0);
35 }
36
37 int insereIni(Deque* dq, int elem){
38     if(dq == NULL) return 0;
39     if(estaCheio(dq)) return 0;
40     dq->ini = (dq->ini-1 < 0 ? MAX-1 :
41         dq->ini-1);
42     dq->dados[dq->ini] = elem;
43     dq->qtd++;
44     return 1;
45 }
46
47 int insereFim(Deque* dq, int elem){
48     if(dq == NULL) return 0;
49     if(estaCheio(dq)) return 0;
50     dq->dados[dq->fim] = elem;
51     dq->fim = (dq->fim+1) % MAX;
52     dq->qtd++;
53     return 1;
54 }
55
56 int removeIni(Deque* dq){
57     if(dq == NULL) return 0;
58     if(estaVazio(dq)) return 0;
59     dq->ini = (dq->ini+1) % MAX;
60     dq->qtd--;
61     return 1;
62 }
63
64 int removeFim(Deque* dq){
65     if(dq == NULL) return 0;
66     if(estaVazio(dq)) return 0;
67     dq->fim = (dq->fim-1 < 0 ? MAX-1 :
68         dq->fim-1);
69     dq->qtd--;
70     return 1;
71 }
72
73 int verInicio(Deque* dq, int* p){
74     if(dq == NULL) return 0;
75     if(estaVazio(dq)) return 0;
76     *p = dq->dados[dq->ini];
77     return 1;
78 }
79
80 int verFim(Deque* dq, int* p){
81     if(dq == NULL) return 0;
82     if(estaVazio(dq)) return 0;
83     *p = dq->dados[dq->fim];
84     return 1;
85 }
86
87 void imprime(Deque* dq){
88     if(dq == NULL) return;
89     if(estaVazio(dq)){
90         printf("Deque Vazio!\n");
91         return;
92     }
93     int i = dq->ini;
94     printf("Elementos: \n");
95     do{
96         printf("%d ", dq->dados[i]);
97         i = (i + 1) % MAX;
98     }while(i != dq->fim);
99     //Usar do..while garante a impressao
100     //de todos elementos
101     //mesmo com a Deque cheia
102     printf("\n");
103 }
104
105 /* Funcao para esperar resposta do
106    usuario e depois limpar*/
107 void aguardaLimpa(){
108     getchar();
109     printf("\n\nAperte qualquer tecla
110     para continuar\n");
111     getchar();
112     system("clear");
113 }

```

codigos/questao11/questao11.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "questao11.h"
4
5  int main(){
6      int escolha_inicial, item;
7      int *inicio =
8          (int*)malloc(sizeof(int));
9      Deque *deque;
10
11     do{
12         printf("0 que deseja fazer?\n1
13         - Criar deque\n2 - Inserir
14         um item no fim\n"
15         "3 - Inserir um item no
16         inicio\n4 - Ver o inicio do
17         Deque;\n5 - Ver o fim do
18         Deque;"
19         "\n6 - Remover um item do
20         fim\n7 - Remover um item do
21         inicio\n"
22         "8 - Imprimir o Deque \n9 -
23         Destruir o Deque \n10 -
24         Sair\n");

```

```

15     scanf("%d",&escolha_inicial);
16
17     switch (escolha_inicial){
18     case 10:
19         break;
20
21     case 1:
22         deque = criaDeque();
23         printf("Deque criado!");
24         aguardaLimpa();
25         break;
26
27     case 2:
28         printf("Digite o numero que
29             deseja adicionar no
30             fim\n");
31         scanf("%d",&item);
32         if(insereFim(deque,item) ==
33             1)
34             printf("Item inserido
35                 no fim!");
36         else
37             printf("Deque cheio ou
38                 inexistente!");
39         aguardaLimpa();
40         break;
41
42     case 3:
43         printf("Digite o numero que
44             deseja adicionar no
45             inicio\n");
46         scanf("%d",&item);
47         if(insereIni(deque,item) ==
48             1)
49             printf("Item inserido
50                 no inicio!");
51         else
52             printf("Deque cheio ou
53                 inexistente!");
54         aguardaLimpa();
55         break;
56
57     case 4:
58         if(verInicio(deque, inicio)
59             == 1)
60             printf("0 valor inicial
61                 do deque eh: %d",
62                 *inicio);
63         else
64             printf("Deque vazio");
65         aguardaLimpa();
66         break;
67
68     case 5:
69         if(verFim(deque, inicio) ==
70             1)
71             printf("0 ultimo valor
72                 do deque eh: %d",
73                 *inicio);
74         else
75             printf("Deque vazio");
76         aguardaLimpa();
77         break;
78
79     case 6:
80         if(removeFim(deque) == 1)
81             printf("Ultimo elemento
82                 do deque removido!");
83         else
84             printf("Deque vazio");
85         aguardaLimpa();
86         break;
87
88     case 7:
89         if(removeIni(deque) == 1)
90             printf("Primeiro
91                 elemento do deque
92                 removido!");
93         else
94             printf("Deque vazio");
95         aguardaLimpa();
96         break;
97
98     case 8:
99         imprime(deque);
100        aguardaLimpa();
101        break;
102
103     case 9:
104        destroiDeque(deque);
105        printf("Deque destruido");
106        aguardaLimpa();
107        break;
108
109     default:
110        printf("Valor invalido");
111        break;
112    }
113    while(escolha_inicial != 10);
114
115    free(inicio);
116
117    return 0;
118 }

```

codigos/questao11/main.c

```

1 all: questao11.o
2   gcc questao11.o main.c -o main
3
4 questao11.o: questao11.h questao11.c
5   gcc -c questao11.c

```

```

6
7 clean:
8   rm -f questao11.o main

```

codigos/questao11/Makefile

Saída

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
1
Deque criado!

Aperte qualquer tecla para continuar

```

Figura 1: Questão 1.1 - Cria Deque

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
3
Digite o numero que deseja adicionar no inicio
5

```

Figura 2: Questão 1.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
2
Digite o numero que deseja adicionar no fim
4
Item inserido no fim!

```

Figura 3: Questão 1.1 - Adiciona Elemento

```

1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
8
Elementos:
5 1 7 4

```

Figura 4: Questão 1.1 - Imprime

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
4
0 valor inicial do deque eh: 5

Aperte qualquer tecla para continuar

```

Figura 5: Questão 1.1 - Ver Início

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
5
0 ultimo valor do deque eh: 4

Aperte qualquer tecla para continuar

```

Figura 6: Questão 1.1 - Ver Fim

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
6
Ultimo elemento do deque removido!

```

Figura 7: Questão 1.1 - Remove Elemento

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
7
Primeiro elemento do deque removido!

```

Figura 8: Questão 1.1 - Remove elemento

```

1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
8
Elementos:
1 7

```

Figura 9: Questão 1.1 - Imprime

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
9
Deque destruido

```

Figura 10: Questão 1.1 - Destrói Deque

1.2 Deque Duplamente Encadeado

Código

```

1  #ifndef DDE_H
2  #define DDE_H
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  typedef struct NO{
8      int info;
9      struct NO* prox;
10     struct NO* ant;
11 }NO;
12
13 typedef struct{
14     int qtd;
15     struct NO* ini;
16     struct NO* fim;
17 }Deque;
18
19 /* Funcoes */
20 NO* alocarNO();

```

```

21 void liberarNO(NO *);
22 Deque* criaDeque();
23 void destroiDeque(Deque *);
24 int tamanhoDeque(Deque *);
25 int estaVazio(Deque *);
26 int insereIni(Deque *, int);
27 int insereFim(Deque *, int);
28 int removeIni(Deque *);
29 int removeFim(Deque *);
30 int verInicio(Deque *, int *);
31 int verFim(Deque *, int *);
32 void imprime(Deque *);
33
34 /*Funcao extra*/
35 void aguardaLimpa();
36
37 #endif

```

codigos/questao12/questao12.h

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "questao12.h"
4
5  NO* alocarNO(){
6      return (NO*) malloc (sizeof(NO));
7  }
8
9  void liberarNO(NO* q){
10     free(q);
11 }
12
13 Deque* criaDeque(){
14     Deque* dq;
15     dq = (Deque*) malloc
16         (sizeof(Deque));
17     if(dq != NULL){
18         dq->qtd = 0;
19         dq->ini = NULL;
20         dq->fim = NULL;
21     }
22     return dq;
23 }
24
25 void destroiDeque(Deque *dq){
26     if(dq != NULL){
27         NO* aux;
28         while(dq->ini != NULL){
29             aux = dq->ini;
30             dq->ini = dq->ini->prox;
31             liberarNO(aux);
32         }
33         free(dq);
34     }
35 }

```

```

36 int tamanhoDeque(Deque *dq){
37     if(dq == NULL)
38         return -1;
39     return dq->qtd;
40 }
41
42 int estaVazio(Deque *dq){
43     if(dq == NULL)
44         return -1;
45     return (dq->qtd == 0);
46 }
47
48 int insereIni(Deque* dq, int elem){
49     if(dq == NULL) return 0;
50     NO* novo = alocarNO();
51     if(novo == NULL) return 0;
52     novo->info = elem;
53     novo->ant = NULL;
54     if(estaVazio(dq)){
55         novo->prox = NULL;
56         dq->fim = novo;
57     }else{
58         dq->ini->ant = novo;
59         novo->prox = dq->ini;
60     }
61     dq->ini = novo;
62     dq->qtd++;
63     return 1;
64 }
65
66 int insereFim(Deque* dq, int elem){
67     if(dq == NULL) return 0;
68     NO* novo = alocarNO();
69     if(novo == NULL) return 0;
70     novo->info = elem;
71     novo->prox = NULL;

```

```

72     if(estaVazio(dq)){
73         novo->ant = NULL;
74         dq->ini = novo;
75     }else{
76         dq->fim->prox = novo;
77         novo->ant = dq->fim;
78     }
79     dq->fim = novo;
80     dq->qtd++;
81     return 1;
82 }
83
84 int removeIni(Deque* dq){
85     if(dq == NULL) return 0;
86     if(estaVazio(dq)) return 0;
87     NO* aux = dq->ini;
88     if(dq->ini == dq->fim){
89         dq->ini = dq->fim = NULL;
90     }else{
91         dq->ini = dq->ini->prox;
92         dq->ini->ant = NULL;
93     }
94     liberarNO(aux);
95     dq->qtd--;
96     return 1;
97 }
98
99 int removeFim(Deque* dq){
100     if(dq == NULL) return 0;
101     if(estaVazio(dq)) return 0;
102     NO* aux = dq->fim;
103     if(dq->ini == dq->fim){
104         dq->ini = dq->fim = NULL;
105     }else{
106         dq->fim = dq->fim->ant;
107         dq->ini->prox = NULL;
108     }
109     liberarNO(aux);
110     dq->qtd--;
111     return 1;
112 }
113

```

```

114 int verInicio(Deque* dq, int* p){
115     if(dq == NULL) return 0;
116     if(estaVazio(dq)) return 0;
117     *p = dq->ini->info;
118     return 1;
119 }
120
121 int verFim(Deque* dq, int* p){
122     if(dq == NULL) return 0;
123     if(estaVazio(dq)) return 0;
124     *p = dq->fim->info;
125     return 1;
126 }
127
128 void imprime(Deque* dq){
129     if(dq == NULL) return;
130     if(estaVazio(dq)){
131         printf("Deque Vazio!\n");
132         return;
133     }
134     NO* aux = dq->ini;
135     printf("Elementos:\n");
136     while(aux != NULL){
137         printf("%d ", aux->info);
138         aux = aux->prox;
139     }
140     printf("\n");
141 }
142
143
144 /* Funcao para esperar resposta do
    usuario e depois limpar*/
145 void aguardaLimpa(){
146     getchar();
147     printf("\n\nAperte qualquer tecla
    para continuar\n");
148     getchar();
149     system("clear");
150 }

```

codigos/questao12/questao12.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "questao12.h"
4
5  int main(){
6      int escolha_inicial, item;
7      int *inicio =
8          (int*)malloc(sizeof(int));
9      Deque *deque;
10
11     do{
12         printf("0 que deseja fazer?\n1
13             - Criar deque\n2 - Inserir
14             um item no fim\n"

```

```

12         "3 - Inserir um item no
13             inicio\n4 - Ver o inicio do
14             Deque;\n5 - Ver o fim do
15             Deque;"
16         "\n6 - Remover um item do
17             fim\n7 - Remover um item do
18             inicio\n"
19         "8 - Imprimir o Deque \n9 -
20             Destruir o Deque \n10 -
21             Sair\n");
22     scanf("%d",&escolha_inicial);
23
24     switch (escolha_inicial){
25     case 10:

```

```

19         break;
20
21     case 1:
22         deque = criaDeque();
23         printf("Deque criado!");
24         aguardaLimpa();
25         break;
26
27     case 2:
28         printf("Digite o numero que
29             deseja adicionar no
30             fim\n");
31         scanf("%d",&item);
32         if(insereFim(deque,item) ==
33             1)
34             printf("Item inserido
35                 no fim!");
36         else
37             printf("Deque cheio ou
38                 inexistente!");
39         aguardaLimpa();
40         break;
41
42     case 3:
43         printf("Digite o numero que
44             deseja adicionar no
45             inicio\n");
46         scanf("%d",&item);
47         if(insereIni(deque,item) ==
48             1)
49             printf("Item inserido
50                 no inicio!");
51         else
52             printf("Deque cheio ou
53                 inexistente!");
54         aguardaLimpa();
55         break;
56     case 4:
57         if(verInicio(deque, inicio)
58             == 1)
59             printf("0 valor inicial
60                 do deque eh: %d",
61                 *inicio);
62         else
63             printf("Deque vazio");
64         aguardaLimpa();
65         break;
66     case 5:
67         if(verFim(deque, inicio) ==
68             1)
69             printf("0 ultimo valor
70                 do deque eh: %d",
71                 *inicio);
72         else
73             printf("Deque vazio");
74         aguardaLimpa();
75         break;
76     case 6:
77         if(removeFim(deque) == 1)
78             printf("Ultimo elemento
79                 do deque removido!");
80         else
81             printf("Deque vazio");
82         aguardaLimpa();
83         break;
84     case 7:
85         if(removeIni(deque) == 1)
86             printf("Primeiro
87                 elemento do deque
88                 removido!");
89         else
90             printf("Deque vazio");
91         aguardaLimpa();
92         break;
93     case 8:
94         imprime(deque);
95         aguardaLimpa();
96         break;
97     case 9:
98         destroiDeque(deque);
99         printf("Deque destruido");
100        aguardaLimpa();
101        break;
102    default:
103        printf("Valor invalido");
104        break;
105    }
106    }while(escolha_inicial != 10);
107    free(inicio);
108    return 0;
109    }

```

codigos/questao12/main.c

```

1 all: questao12.o
2 gcc questao12.o main.c -o main
3
4 questao12.o: questao12.h questao12.c
5 gcc -c questao12.c

```

```

6
7 clean:
8 rm -f questao12.o main
9
10 codigos/questao12/Makefile

```


Saída

```
0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
1
Deque criado!

Aperte qualquer tecla para continuar
```

Figura 11: Questão 1.2 - Cria Deque

```
0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
2
Digite o numero que deseja adicionar no fim
1
Item inserido no fim!
```

Figura 12: Questão 1.2 - Adiciona Elemento

```
0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
3
Digite o numero que deseja adicionar no inicio
6
Item inserido no inicio!
```

Figura 13: Questão 1.2 - Adiciona Elemento

```
0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
8
Elementos:
6 3 8 1
```

Figura 14: Questão 1.2 - Imprime

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
4
0 valor inicial do deque eh: 6
Aperte qualquer tecla para continuar

```

Figura 15: Questão 1.2 - Ver Início

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
5
0 ultimo valor do deque eh: 1
Aperte qualquer tecla para continuar

```

Figura 16: Questão 1.2 - Ver Fim

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
6
Ultimo elemento do deque removido!

```

Figura 17: Questão 1.2 - Remove Elemento

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
7
Primeiro elemento do deque removido!

```

Figura 18: Questão 1.2 - Remove elemento

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
8
Elementos:
3 8

```

Figura 19: Questão 1.2 - Imprime

```

0 que deseja fazer?
1 - Criar deque
2 - Inserir um item no fim
3 - Inserir um item no inicio
4 - Ver o inicio do Deque;
5 - Ver o fim do Deque;
6 - Remover um item do fim
7 - Remover um item do inicio
8 - Imprimir o Deque
9 - Destruir o Deque
10 - Sair
9
Deque destruido

```

Figura 20: Questão 1.2 - Destrói Deque

2 Fila de Prioridades

2.1 Fila de Prioridades Simplesmente Encadeada

Código

```

1 #ifndef FPSE_H
2 #define FPSE_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 typedef struct NO{
8     int info, prio;
9     struct NO* prox;
10 }NO;
11
12 typedef struct NO* Filap;
13
14 /*Funcoes*/
15 NO* alocarNO();

```

```

16 void liberarNO(NO *);
17 Filap* criaFila();
18 int estaVazia(Filap *);
19 int inserirPrio(Filap *, int, int);
20 int removeIni(Filap *);
21 int verIni(Filap *, int *);
22 void imprime(Filap *);
23 void destroiFila(Filap *);
24
25 /*Funcao extra*/
26 int conta(Filap *);
27 void aguardaLimpa();
28 #endif

```

codigos/questao21/questao21.h

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao21.h"
4
5 NO* alocarNO(){
6     return (NO*) malloc (sizeof(NO));
7 }
8
9 void liberarNO(NO* q){
10     free(q);
11 }
12
13 Filap* criaFila(){
14     Filap* fp;
15     fp = (Filap*) malloc
16         (sizeof(Filap));
17     if(fp != NULL)
18         *fp = NULL;
19     return fp;
20 }
21
22 int estaVazia(Filap* fp){
23     if(fp == NULL) return -1;
24     return ((*fp) == NULL);
25 }
26
27 int inserirPrio(Filap* fp, int elem,
28     int pri){
29     if(fp == NULL) return 0;
30     NO* novo = alocarNO();
31     if(novo == NULL) return 0;
32
33     novo->info = elem;
34     novo->prio = pri;
35
36     if(estaVazia(fp)){
37         novo->prox = *fp;
38         *fp = novo;
39     }else{
40         NO* aux, *ant;

```

```

39     ant = NULL;
40     aux = *fp; //Inicio
41     while(aux != NULL && aux->prio
42         >= novo->prio){
43         ant = aux;
44         aux = aux->prox;
45     }
46     if(ant == NULL){
47         novo->prox = *fp;
48         *fp = novo;
49     }else{
50         novo->prox = ant->prox;
51         ant->prox = novo;
52     }
53     return 1;
54 }
55
56 int removeIni(Filap* fp){
57     if(fp == NULL) return 0;
58     if(estaVazia(fp)) return 0;
59     NO* aux = *fp;
60     *fp = aux->prox;
61     liberarNO(aux);
62     return 1;
63 }
64
65 int verIni(Filap* fp, int* p){
66     if(fp == NULL) return 0;
67     if(estaVazia(fp)) return 0;
68     *p = (*fp)->info;
69     return 1;
70 }
71
72 void imprime(Filap* fp){
73     if(fp == NULL) return;
74     if(estaVazia(fp)){
75         printf("Fila Vazia!\n");
76         return;
77     }

```

```

78     NO* aux = *fp;
79     while(aux != NULL){
80         printf("[%d, %d] ", aux->prio,
81             aux->info);
82         aux = aux->prox;
83     }
84     printf("\n");
85 }
86 void destroiFila(FilaP* fp){
87     if(fp != NULL){
88         NO* aux;
89         while((*fp) != NULL){
90             aux = *fp;
91             *fp = (*fp)->prox;
92             liberarNO(aux);
93         }
94         free(fp);
95     }
96 }
97
98 /* Funcao que conta os itens na fila de
99    prioridade*/
100 int conta(FilaP* fp){
101     if(fp == NULL) return 0;
102     if(estaVazia(fp)){
103         printf("Fila Vazia!\n");
104         return 0;
105     }
106     int contador = 0;
107
108     NO* aux = *fp;
109     while(aux != NULL){
110         contador++;
111         aux = aux->prox;
112     }
113     return contador;
114 }
115 }
116
117 /* Funcao para esperar resposta do
118    usuario e depois limpar*/
119 void aguardaLimpa(){
120     getchar();
121     printf("\n\nAperte qualquer tecla
122         para continuar\n");
123     getchar();
124     system("clear");
125 }

```

codigos/questao21/questao21.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "questao21.h"
4
5  int main(){
6      int escolha_inicial, item,
7          prioridade;
8      int *inicio =
9          (int*)malloc(sizeof(int));
10     FilaP *filap;
11     do{
12         printf("0 que deseja fazer?\n1
13             - Criar Fila\n2 - Inserir um
14             item pela prioridade\n"
15             "3 - Ver o inicio da Fila\n4 -
16             Remover um item;\n5 -
17             Imprimir a Fila;"
18             "\n6 - Mostrar o tamanho da
19             Fila\n7 - Destruir a Fila\n8
20             - Sair\n");
21         scanf("%d",&escolha_inicial);
22
23         switch (escolha_inicial){
24             case 8:
25                 break;
26             case 1:
27                 filap = criaFila();
28
29                 printf("Fila criada!");
30                 aguardaLimpa();
31                 break;
32             case 2:
33                 printf("Digite o numero que
34                     deseja adicionar, junto
35                     a sua prioridade\n");
36                 scanf("%d
37                     %d",&item,&prioridade);
38                 if(inserirPrio(filap,item,
39                     prioridade) == 1)
40                     printf("Item inserido
41                         na sua prioridade!");
42                 else
43                     printf("Fila cheia ou
44                         inexistente!");
45                 aguardaLimpa();
46                 break;
47             case 3:
48                 if(verIni(filap, inicio) ==
49                     1)
50                     printf("0 valor inicial
51                         da fila eh: %d",
52                         *inicio);
53                 else
54                     printf("Fila vazia");
55                 aguardaLimpa();

```

```

42         break;
43
44     case 4:
45         if(removeIni(filap) == 1)
46             printf("Elemento de
                     maior prioridade da
                     fila foi removido!");
47         else
48             printf("Fila vazio");
49         aguardaLimpa();
50         break;
51
52     case 5:
53         imprime(filap);
54         aguardaLimpa();
55         break;
56
57     case 6:
58         printf("O tamanho da Fila
                 de Prioridade eh %d",
                 conta(filap));
59         aguardaLimpa();
60         break;
61
62     case 7:
63         destroiFila(filap);
64         printf("Fila destruido");
65         aguardaLimpa();
66         break;
67
68     default:
69         printf("Valor invalido");
70         break;
71     }
72 }while(escolha_inicial != 8);
73
74 free(inicio);
75
76 return 0;
77 }

```

codigos/questao21/main.c

```

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

```

codigos/questao21/Makefile

Saída

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
1
Fila criada!

Aperte qualquer tecla para continuar

```

Figura 21: Questão 2.1 - Cria Deque

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
2
Digite o numero que deseja adicionar, junto a sua prioridade
4
5
Item inserido na sua prioridade!

Aperte qualquer tecla para continuar

```

Figura 22: Questão 2.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
2
Digite o numero que deseja adicionar, junto a sua prioridade
1
9
Item inserido na sua prioridade!
Aperte qualquer tecla para continuar

```

Figura 23: Questão 2.1 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
5
[9, 1] [5, 4]

```

Figura 24: Questão 2.1 - Imprime

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
3
0 valor inicial da fila eh: 1
Aperte qualquer tecla para continuar

```

Figura 25: Questão 2.1 - Ver Início

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
4
Elemento de maior prioridade da fila foi removido!

```

Figura 26: Questão 2.1 - Remove Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
6
0 tamanho da Fila de Prioridade eh 1

```

Figura 27: Questão 2.1 - Tamanho do Deque

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
7
Fila destruido

```

Figura 28: Questão 2.1 - Destrói Deque

2.2 Fila de Prioridades com Heap Binária

Código

```

1 #ifndef FPHEAP_H
2 #define FPHEAP_H
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 #define MAX 100

```

```

8 typedef struct NO{
9     int info, prio;
10 }NO;
11 typedef struct{
12     int qtd;
13     NO dados[MAX];
14 }FilaP;

```

```

15
16 /* Funcoes */
17 Filap* criaFila();
18 void destroiFila(Filap *);
19 int tamanhoFila(Filap *);
20 int estaCheia(Filap *);
21 int estaVazia(Filap *);
22 void trocaNO(NO *, NO *);
23 void ajustaHeapInsere(Filap *, int);
24 int inserirPrio(Filap *, int, int);
25 void ajustaHeapRemove(Filap *, int);

```

```

26 int removeIni(Filap *);
27 int verIni(Filap *, int *, int *);
28 void imprime(Filap *);
29
30 /*Funcao extra*/
31 void aguardaLimpa();
32
33 #endif

```

codigos/questao22/questao22.h

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao22.h"
4
5 Filap* criaFila(){
6     Filap* fp;
7     fp = (Filap*) malloc
8         (sizeof(Filap));
9     if(fp != NULL)
10         fp->qtd = 0;
11     return fp;
12 }
13 void destroiFila(Filap* fp){
14     if(fp != NULL)
15         free(fp);
16 }
17
18 int tamanhoFila(Filap *fp){
19     if(fp == NULL) return -1;
20     return fp->qtd;
21 }
22
23 int estaCheia(Filap *fp){
24     if(fp == NULL) return -1;
25     return (fp->qtd == MAX);
26 }
27
28 int estaVazia(Filap *fp){
29     if(fp == NULL) return -1;
30     return (fp->qtd == 0);
31 }
32
33 void trocaNO(NO* a, NO* b){
34     NO temp;
35     temp.info = a->info;
36     temp.prio = a->prio;
37     a->info = b->info;
38     a->prio = b->prio;
39     b->info = temp.info;
40     b->prio = temp.prio;
41 }
42
43 void ajustaHeapInsere(Filap* fp, int
    filho){

```

```

44     NO temp;
45     int pai = (filho-1)/2;
46     int prioPai = fp->dados[pai].prio;
47     int prioFilho =
48         fp->dados[filho].prio;
49     while(filho > 0 && prioPai <
50         prioFilho){
51         trocaNO(&fp->dados[filho],
52             &fp->dados[pai]);
53         filho = pai;
54         pai = (pai-1)/2;
55         prioPai = fp->dados[pai].prio;
56         prioFilho =
57             fp->dados[filho].prio;
58     }
59 }
60
61 int inserirPrio(Filap* fp, int elem,
62     int pri){
63     if(fp == NULL) return 0;
64     if(estaCheia(fp)) return 0;
65     fp->dados[fp->qtd].info = elem;
66     fp->dados[fp->qtd].prio = pri;
67     ajustaHeapInsere(fp, fp->qtd);
68     fp->qtd++;
69     return 1;
70 }
71
72 void ajustaHeapRemove(Filap* fp, int
73     pai){
74     NO temp;
75     int filho = 2*pai + 1;
76     while(filho < fp->qtd){
77         if(filho < fp->qtd-1)
78             if(fp->dados[filho].prio <
79                 fp->dados[filho+1].prio)
80                 filho++;
81
82         if(fp->dados[pai].prio >
83             fp->dados[filho].prio)
84             break;
85
86         trocaNO(&fp->dados[pai],
87             &fp->dados[filho]);
88         pai = filho;

```

```

80     filho = 2*pai + 1;
81 }
82 }
83
84 int removeIni(FilaP* fp){
85     if(fp == NULL) return 0;
86     if(estaVazia(fp)) return 0;
87
88     fp->qtd--;
89     fp->dados[0].info =
90         fp->dados[fp->qtd].info;
91     fp->dados[0].prio =
92         fp->dados[fp->qtd].prio;
93     ajustaHeapRemove(fp, 0);
94     return 1;
95 }
96
97 int verIni(FilaP* fp, int* valor, int*
98     pri){
99     if(fp == NULL) return 0;
100     if(estaVazia(fp)) return 0;
101     *valor = fp->dados[0].info;
102     *pri = fp->dados[0].prio;
103     return 1;
104 }
105
106 void imprime(FilaP* fp){
107     if(fp == NULL) return;
108     if(estaVazia(fp)){
109         printf("Fila Vazia!\n");
110         return;
111     }
112     printf("Elementos:\n");
113     int i;
114     for(i=0; i<fp->qtd; i++)
115         printf("[%d, %d] (%d) -- ",
116             fp->dados[i].prio,
117             fp->dados[i].info, i);
118     printf("\n");
119 }
120
121 /* Funcao para esperar resposta do
122    usuario e depois limpar*/
123 void aguardaLimpa(){
124     getchar();
125     printf("\n\nAperte qualquer tecla
126         para continuar\n");
127     getchar();
128     system("clear");
129 }

```

codigos/questao22/questao22.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao22.h"
4
5 int main(){
6     int escolha_inicial, item,
7     prioridade;
8     int *inicio =
9         (int*)malloc(sizeof(int));
10    int *prio_inicio =
11        (int*)malloc(sizeof(int));
12    FilaP *filap;
13
14    do{
15        printf("0 que deseja fazer?\n1
16            - Criar Fila\n2 - Inserir um
17            item pela prioridade\n"
18            "3 - Ver o inicio da Fila\n4 -
19            Remover um item;\n5 -
20            Imprimir a Fila;"
21            "\n6 - Mostrar o tamanho da
22            Fila\n7 - Destruir a Fila\n8
23            - Sair\n");
24        scanf("%d",&escolha_inicial);
25
26        switch (escolha_inicial){
27            case 8:
28                break;
29
30            case 1:
31                filap = criaFila();
32                printf("Fila criada!");
33                aguardaLimpa();
34                break;
35
36            case 2:
37                printf("Digite o numero que
38                deseja adicionar, junto
39                a sua prioridade\n");
40                scanf("%d", &item, &prioridade);
41                if(inserirPrio(filap, item,
42                    prioridade) == 1)
43                    printf("Item inserido
44                    na sua prioridade!");
45                else
46                    printf("Fila cheia ou
47                    inexistente!");
48                aguardaLimpa();
49                break;
50
51            case 3:
52                if(verIni(filap, inicio,
53                    prio_inicio) == 1)
54                    printf("0 valor inicial
55                    da fila e sua
56                    prioridade eh:
57                    (%d,%d)",

```



```

40         *inicio, *prio_inicio);
41     else
42         printf("Fila vazia");
43     aguardaLimpa();
44     break;
45
46     case 4:
47         if(removeIni(filap) == 1)
48             printf("Elemento de
49                 maior prioridade da
50                 fila foi removido!");
51         else
52             printf("Fila vazia");
53         aguardaLimpa();
54         break;
55
56     case 5:
57         imprime(filap);
58         aguardaLimpa();
59         break;
60
61     case 6:
62         printf("O tamanho da Fila
63
64         de Prioridade eh %d",
65             filap->qtd);
66         aguardaLimpa();
67         break;
68
69     case 7:
70         destroiFila(filap);
71         printf("Fila destruido");
72         aguardaLimpa();
73         break;
74
75     default:
76         printf("Valor invalido");
77         break;
78 }
79 while(escolha_inicial != 8);
80
81 free(inicio);
82
83 return 0;
84 }

```

codigos/questao22/main.c

```

1 all: questao22.o
2     gcc questao22.o main.c -o main
3
4 questao22.o: questao22.h questao22.c
5     gcc -c questao22.c
6
7 clean:
8     rm -f questao22.o main

```

codigos/questao22/Makefile

Saída

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
1
Fila criada!

Aperte qualquer tecla para continuar

```

Figura 29: Questão 2.2 - Cria Deque

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
2
Digite o numero que deseja adicionar, junto a sua prioridade
7
Item inserido na sua prioridade!

Aperte qualquer tecla para continuar

```

Figura 30: Questão 2.2 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
2
Digite o numero que deseja adicionar, junto a sua prioridade
3
4
Item inserido na sua prioridade!
Aperte qualquer tecla para continuar

```

Figura 31: Questão 2.2 - Adiciona Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
5
Elementos:
[4, 3] (0) -- [2, 7] (1) --
Aperte qualquer tecla para continuar

```

Figura 32: Questão 2.2 - Imprime

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
3
0 valor inicial da fila e sua prioridade eh: (3,4)
Aperte qualquer tecla para continuar

```

Figura 33: Questão 2.2 - Ver Início

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
4
Elemento de maior prioridade da fila foi removido!
Aperte qualquer tecla para continuar

```

Figura 34: Questão 2.2 - Remove Elemento

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
6
0 tamanho da Fila de Prioridade eh 1
Aperte qualquer tecla para continuar

```

Figura 35: Questão 2.2 - Tamanho do Deque

```

0 que deseja fazer?
1 - Criar Fila
2 - Inserir um item pela prioridade
3 - Ver o inicio da Fila
4 - Remover um item;
5 - Imprimir a Fila;
6 - Mostrar o tamanho da Fila
7 - Destruir a Fila
8 - Sair
7
Fila destruido
Aperte qualquer tecla para continuar

```

Figura 36: Questão 2.2 - Destrói Deque