

Lista 5 de Laboratório de Programação II

Nome : Messias Feres Curi Melo

Matrícula : 2022003764

Problema 1


1.1

main.c

```
C: > Users > Messi > OneDrive > Área de Trabalho > C main.c > main()

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include "fila.h"
5
6  int main()
7  {
8      Fila *F;
9      int elem, escolha;
10     printf("-<X Menu X>-\n");
11     printf("1 - Criar fila\n");
12     printf("2 - Enfileirar um item\n");
13     printf("3 - Ver o início da fila\n");
14     printf("4 - Desenfileirar um item\n");
15     printf("5 - Imprimir a fila\n");
16     printf("6 - Destruir a fila\n");
17     printf("7 - Sair\n");
18
19     while(true){
20         printf("Escolha: ");
21         scanf("%d", &escolha);
22
23         if(escolha == 1){
24             F = criaFila();
25             printf("-> Criado com sucesso!\n");
26         }else if(escolha == 2){
27             printf("-> Elemento a ser enfileirado: ");
28             scanf("%d", &elem);
29             enfileirar(F, elem);
30             printf("-> Enfileirado com sucesso!\n");
31         }else if(escolha == 3){
32             verInicio(F, &elem);
33             printf("-> Frente: %d\n", elem);
34         }else if(escolha == 4){
35             printf("-> Desenfileirado primeiro elemento\n");
36             desenfileirar(F);
37             printf("-> Desenfileirado com sucesso!\n");
38         }else if(escolha == 5){
39             imprime(F);
40         }else if(escolha == 6){
41             destroiFila(F);
42             printf("-> Destruído com sucesso!\n");
43         }else if(escolha == 7){
44             printf("-> Finalizando!\n");
45             break;
46         }else{
47             printf("-> Número inválido!\n");
48         }
49     }
50     return 0;
51 }
```

fila.c

C: > Users > Messi > OneDrive > Área de Trabalho > C fila.c >  imprime(Fila *)

```
1  #include "fila.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  Fila* criaFila(){
6      Fila* fi;
7      fi = (Fila*) malloc (sizeof(Fila));
8      if(fi != NULL){
9          fi->qtd = fi->ini = fi->fim = 0;
10     }
11     return fi;
12 }
13
14 void destroiFila(Fila *fi){
15     if(fi != NULL)
16         free(fi);
17 }
18
19 int tamanhoFila(Fila *fi){
20     if(fi == NULL)
21         return -1;
22     return fi->qtd;
23 }
24
25 int estaCheia(Fila *fi){
26     if(fi == NULL)
27         return -1;
28     return (fi->qtd == MAX);
29 }
30
31 int estaVazia(Fila *fi){
32     if(fi == NULL)
33         return -1;
34     return (fi->qtd == 0);
35 }
36
37 int enqueue(Fila* fi, int elem){
38     if(fi == NULL) return 0;
39     if(estaCheia(fi)) return 0;
40     fi->dados[fi->fim] = elem;
41     fi->fim = (fi->fim+1) % MAX;
42     fi->qtd++;
43     return 1;
44 }
45
46 int dequeue(Fila* fi){
47     if(fi == NULL) return 0;
48     if(estaVazia(fi)) return 0;
49     fi->ini = (fi->ini+1) % MAX;
50     fi->qtd--;
51     return 1;
52 }
53
54 int verInicio(Fila* fi, int* p){
55     if(fi == NULL) return 0;
56     if(estaVazia(fi)) return 0;
57     *p = fi->dados[fi->ini];
58     return 1;
59 }
60
61 void imprime(Fila* fi){
62     if(fi == NULL) return;
63     if(estaVazia(fi)){
64         printf("Fila Vazia!\n");
65         return;
66     }
67     int i = fi->ini;
68     printf("Elementos: \n");
69     do{
70         printf("%d ", fi->dados[i]);
71         i = (i + 1) % MAX;
72     }while(i != fi->fim);
73     printf("\n");
74 }
75
```

fila.h

```
C:\Users\Messi\OneDrive\Área de Trabalho > C:\fila.h > MAX
1  #ifndef FILA_H
2  #define FILA_H
3
4  #define MAX 100
5
6  typedef struct{
7      int qtd, ini, fim;
8      int dados[MAX];
9  }Fila;
10
11  Fila* criaFila();
12
13  void destroiFila(Fila *fi);
14
15  int tamanhoFila(Fila *fi);
16
17  int estaCheia(Fila *fi);
18
19  int estaVazia(Fila *fi);
20
21  int enfileirar(Fila* fi, int elem);
22
23  int desenfileirar(Fila* fi);
24
25  int verInicio(Fila* fi, int* p);
26
27  void imprime(Fila* fi);
28
29  #endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$ gcc main.c fila.h fila.c -o f1
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$ ./f1
-cX Menu X>-
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
Escolha: 1
-> Criado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 10
-> Efileirado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 13
-> Efileirado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 2
-> Efileirado com sucesso!
Escolha: 3
-> Frente: 10
Escolha: 4
-> Desenfileirado primeiro elemento
-> Desenfileirado com sucesso!
Escolha: 3
-> Frente: 13
Escolha: 5
Elementos:
13 2
Escolha: 6
-> Destruído com sucesso!
Escolha: 7
-> Finalizando!
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$
```

1.2

main.c

```
C main.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include "fila.h"
5
6  int main(){
7      Fila *F;
8      int elem, escolha;
9      printf("-<X Menu X>-\n");
10     printf("1 - Criar fila\n");
11     printf("2 - Enfileirar um item\n");
12     printf("3 - Ver o início da fila\n");
13     printf("4 - Desenfileirar um item\n");
14     printf("5 - Imprimir a fila\n");
15     printf("6 - Destruir a fila\n");
16     printf("7 - Sair\n");
17
18     while(true){
19         printf("Escolha: ");
20         scanf("%d", &escolha);
21
22         if(escolha == 1){
23             F = criaFila();
24             printf("-> Criado com sucesso!\n");
25         }else if(escolha == 2){
26             printf("-> Elemento a ser enfileirado: ");
27             scanf("%d",&elem);
28             enfileirar(F, elem);
29             printf("-> Enfileirado com sucesso!\n");
30         }else if(escolha == 3){
31             verInicio(F, &elem);
32             printf("-> Frente: %d\n", elem);
33         }else if(escolha == 4){
34             printf("-> Desenfileirado primeiro elemento\n");
35             desenfileirar(F);
36             printf("-> Desenfileirado com sucesso!\n");
37         }else if(escolha == 5){
38             imprime(F);
39         }else if(escolha == 6){
40             destroiFila(F);
41             printf("-> Destruído com sucesso!\n");
42         }else if(escolha == 7){
43             printf("-> Finalizando!\n");
44             break;
45         }else{
46             printf("-> Número inválido!\n");
47         }
48     }
49     return 0;
50 }
51
```

fila.c

```
C fila.c > ❏ destroiFila(Fila *)
1  #include "fila.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  Fila* criaFila(){
6      Fila* fi = (Fila*)malloc(sizeof(Fila));
7      if (fi != NULL){
8          fi->qtd = 0;
9          fi->ini = NULL;
10         fi->fim = NULL;
11     }
12     return fi;
13 }
14
15 void destroiFila(Fila* fi){
16     if (fi != NULL){
17         NO* aux;
18         while (fi->ini != NULL){
19             aux = fi->ini;
20             fi->ini = fi->ini->prox;
21             free(aux);
22         }
23         free(fi);
24     }
25 }
26
27 int tamanhoFila(Fila* fi){
28     if (fi == NULL)
29         return -1;
30     return fi->qtd;
31 }
32
33 int estaVazia(Fila* fi){
34     if (fi == NULL)
35         return -1;
36     return (fi->qtd == 0);
37 }
38
39 int enfileirar(Fila* fi, int elem){
40     if (fi == NULL)
41         return 0;
42     NO* novo = (NO*)malloc(sizeof(NO));
43     if (novo == NULL)
44         return 0;
45     novo->info = elem;
46     novo->prox = NULL;
47     if (fi->fim == NULL){
```

```

48     fi->ini = novo;
49 } else{
50     fi->fim->prox = novo;
51 }
52 fi->fim = novo;
53 fi->qtd++;
54 return 1;
55 }
56
57 int desenfileirar(Fila* fi){
58     if (fi == NULL || fi->ini == NULL)
59         return 0;
60     NO* aux = fi->ini;
61     fi->ini = fi->ini->prox;
62     free(aux);
63     if (fi->ini == NULL)
64         fi->fim = NULL;
65     fi->qtd--;
66     return 1;
67 }
68
69 int verInicio(Fila* fi, int* p){
70     if (fi == NULL || fi->ini == NULL)
71         return 0;
72     *p = fi->ini->info;
73     return 1;
74 }
75
76 void imprime(Fila* fi){
77     if (fi == NULL) return;
78     if (estaVazia(fi)){
79         printf("Fila Vazia!\n");
80         return;
81     }
82     NO* atual = fi->ini;
83     printf("Elementos: \n");
84     while (atual != NULL){
85         printf("%d ", atual->info);
86         atual = atual->prox;
87     }
88     printf("\n");
89 }

```

fila.h

```

C: filah > ...
1  #ifndef FILA_H
2  #define FILA_H
3
4  typedef struct NO {
5      int info;
6      struct NO* prox;
7  } NO;
8
9  typedef struct {
10     int qtd;
11     NO* ini;
12     NO* fim;
13 } Fila;
14
15 Fila* criaFila();
16
17 void destroiFila(Fila *fi);
18
19 int tamanhoFila(Fila *fi);
20
21 int estaCheia(Fila *fi);
22
23 int estaVazia(Fila *fi);
24
25 int enfileirar(Fila* fi, int elem);
26
27 int desenfileirar(Fila* fi);
28
29 int verInicio(Fila* fi, int* p);
30
31 void imprime(Fila* fi);
32
33 #endif
34

```

Console

```
messiasfcn@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/f2$ gcc main.c fila.h fila.c -o f2
messiasfcn@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/f2$ ./f2
-<X Menu X>-
1 - Criar fila
2 - Enfileirar um item
3 - Ver o início da fila
4 - Desenfileirar um item
5 - Imprimir a fila
6 - Destruir a fila
7 - Sair
Escolha: 1
-> Criado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 10
-> Enfileirado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 17
-> Enfileirado com sucesso!
Escolha: 2
-> Elemento a ser enfileirado: 4
-> Enfileirado com sucesso!
Escolha: 3
-> Frente: 10
Escolha: 4
-> Desenfileirado primeiro elemento
-> Desenfileirado com sucesso!
Escolha: 17
-> Número inválido!
Escolha: 5
Elementos:
17 4
Escolha: 6
-> Destruído com sucesso!
Escolha: 7
-> Finalizando!
messiasfcn@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/f2$
```

Problema 2

2.1

main.c

```
C main.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include "pilha.h"
5
6  int main(){
7      Pilha *P;
8      int elem, escolha;
9      printf("-<X Menu X>-\n");
10     printf("1 - Criar pilha\n");
11     printf("2 - Empilhar um item\n");
12     printf("3 - Ver o topo da pilha\n");
13     printf("4 - Desempilhar um item\n");
14     printf("5 - Imprimir a pilha\n");
15     printf("6 - Destruir a pilha\n");
16     printf("7 - Sair\n");
17
18     while(true){
19         printf("Escolha: ");
20         scanf("%d", &escolha);
21
22         if(escolha == 1){
23             P = criaPilha();
24             printf("-> Criado com sucesso!\n");
25         }else if(escolha == 2){
26             printf("-> Elemento a ser empilhado: ");
27             scanf("%d",&elem);
28             empilhar(P, elem);
29             printf("-> Empilhado com sucesso!\n");
30         }else if(escolha == 3){
31             verTopo(P, &elem);
32             printf("-> O topo da pilha é : %d\n", elem);
33         }else if(escolha == 4){
34             desempilhar(P);
35             printf("-> Desempilhado com sucesso!\n");
36         }else if(escolha == 5){
37             imprime(P);
38         }else if(escolha == 6){
39             destroiPilha(P);
40             printf("-> Destruído com sucesso!\n");
41         }else if(escolha == 7){
42             printf("-> Finalizando!\n");
43             break;
44         }else{
45             printf("-> Número inválido!\n");
46         }
47     }
48
49     return 0;
50 }
```


pilha.c

```
C pilha.c > tamanhoPilha(Pilha *)
1  #include "pilha.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  Pilha* criaPilha(){
6      Pilha* pi;
7      pi = (Pilha*) malloc (sizeof(Pilha));
8      if(pi != NULL){
9          pi->topo = 0;
10     }
11     return pi;
12 }
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     }
```

pilha.h

```
C pilha.h > ...
1  #ifndef PILHA_H
2  #define PILHA_H
3
4  #define MAX 100
5
6  typedef struct{
7      int topo;
8      int dados[MAX];
9  }Pilha;
10
11  Pilha* criaPilha();
12
13  void destroiPilha(Pilha *pi);
14
15  int tamanhoPilha(Pilha *pi);
16
17  int estaCheia(Pilha *pi);
18
19  int estaVazia(Pilha *pi);
20
21  int empilhar(Pilha* pi, int elem);
22
23  int desempilhar(Pilha* pi);
24
25  int verTopo(Pilha* pi, int* p);
26
27  void imprime(Pilha* pi);
28
29  #endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p1$ gcc main.c pilha.h pilha.c -o p1
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p1$ ./p1
-<X Menu X>-
1 - Criar pilha
2 - Empilhar um item
3 - Ver o topo da pilha
4 - Desempilhar um item
5 - Imprimir a pilha
6 - Destruir a pilha
7 - Sair
Escolha: 1
-> Criado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 10
-> Empilhado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 17
-> Empilhado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 5
-> Empilhado com sucesso!
Escolha: 3
-> O topo da pilha é : 5
Escolha: 4
-> Desempilhado com sucesso!
Escolha: 3
-> O topo da pilha é : 17
Escolha: 5
Elementos:
17 10
Escolha: 4
-> Desempilhado com sucesso!
Escolha: 5
Elementos:
10
Escolha: 6
-> Destruido com sucesso!
Escolha: 7
-> Finalizando!
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p1$
```

2.2

main.c

```
C main.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include "pilha.h"
5
6  int main(){
7      Pilha *P;
8      int elem, escolha;
9      printf("-<X Menu X>-\n");
10     printf("1 - Criar pilha\n");
11     printf("2 - Empilhar um item\n");
12     printf("3 - Ver o topo da pilha\n");
13     printf("4 - Desempilhar um item\n");
14     printf("5 - Imprimir a pilha\n");
15     printf("6 - Destruir a pilha\n");
16     printf("7 - Sair\n");
17
18     while(true){
19         printf("Escolha: ");
20         scanf("%d", &escolha);
21
22         if(escolha == 1){
23             P = criaPilha();
24             printf("-> Criado com sucesso!\n");
25         }else if(escolha == 2){
26             printf("-> Elemento a ser empilhado: ");
27             scanf("%d",&elem);
28             empilhar(P, elem);
29             printf("-> Empilhado com sucesso!\n");
30         }else if(escolha == 3){
31             verTopo(P, &elem);
32             printf("-> O topo da pilha é : %d\n", elem);
33         }else if(escolha == 4){
34             desempilhar(P);
35             printf("-> Desempilhado com sucesso!\n");
36         }else if(escolha == 5){
37             imprime(P);
38         }else if(escolha == 6){
39             destroiPilha(P);
40             printf("-> Destruído com sucesso!\n");
41         }else if(escolha == 7){
42             printf("-> Finalizando!\n");
43             break;
44         }else{
45             printf("-> Número inválido!\n");
46         }
47     }
48
49     return 0;
50 }
```

pilha.c

```
C pilha.c > empilhar(Pilha *, int)
1  #include "pilha.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  Pilha* criaPilha() {
6      Pilha* pi = (Pilha*)malloc(sizeof(Pilha));
7      if (pi != NULL) {
8          pi->qtd = 0;
9          pi->topo = NULL;
10     }
11     return pi;
12 }
13
14 void destroiPilha(Pilha* pi) {
15     if (pi != NULL) {
16         while (pi->topo != NULL) {
17             NO* aux = pi->topo;
18             pi->topo = pi->topo->prox;
19             free(aux);
20         }
21         free(pi);
22     }
23 }
24
25 int tamanhoPilha(Pilha* pi) {
26     if (pi == NULL)
27         return -1;
28     return pi->qtd;
29 }
30
31 int estaVazia(Pilha* pi) {
32     if (pi == NULL)
33         return -1;
34     return (pi->topo == NULL);
35 }
36
37 int empilhar(Pilha* pi, int elem) {
38     if (pi == NULL)
39         return 0;
40
41     NO* novo_no = (NO*)malloc(sizeof(NO));
42     if (novo_no == NULL)
43         return 0;
44
45     novo_no->info = elem;
46     novo_no->prox = pi->topo;
47     pi->topo = novo_no;
48
49     pi->qtd++;
50     return 1;
51 }
52
53 int desempilhar(Pilha* pi) {
54     if (pi == NULL || pi->topo == NULL)
55         return 0;
56
57     NO* aux = pi->topo;
58     pi->topo = aux->prox;
59     free(aux);
60     pi->qtd--;
61     return 1;
62 }
63
64 int verTopo(Pilha* pi, int* p) {
65     if (pi == NULL || pi->topo == NULL)
66         return 0;
67
68     *p = pi->topo->info;
69     return 1;
70 }
71
72 void imprime(Pilha* pi) {
73     if (pi == NULL) return;
74
75     NO* aux = pi->topo;
76     printf("Elementos: ");
77     while (aux != NULL) {
78         printf("%d ", aux->info);
79         aux = aux->prox;
80     }
81     printf("\n");
82 }
```

pilha.h

```
C pilha > ...
1  ~ #ifndef PILHA_H
2    #define PILHA_H
3
4  ~ typedef struct NO {
5      int info;
6      struct NO* prox;
7  } NO;
8
9  ~ typedef struct {
10     int qtd;
11     NO* topo;
12 } Pilha;
13
14 Pilha* criaPilha();
15
16 void destroiPilha(Pilha* pi);
17
18 int tamanhoPilha(Pilha* pi);
19
20 int estaVazia(Pilha* pi);
21
22 int empilhar(Pilha* pi, int elem);
23
24 int desempilhar(Pilha* pi);
25
26 int verTopo(Pilha* pi, int* p);
27
28 void imprime(Pilha* pi);
29
30 #endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p2$ gcc main.c pilha.h pilha.c -o p2
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p2$ ./p2
-X Menu X-
1 - Criar pilha
2 - Empilhar um item
3 - Ver o topo da pilha
4 - Desempilhar um item
5 - Imprimir a pilha
6 - Destruir a pilha
7 - Sair
Escolha: 1
-> Criado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 12
-> Empilhado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 17
-> Empilhado com sucesso!
Escolha: 2
-> Elemento a ser empilhado: 3
-> Empilhado com sucesso!
Escolha: 3
-> O topo da pilha é : 3
Escolha: 4
-> Desempilhado com sucesso!
Escolha: 3
-> O topo da pilha é : 17
Escolha: 5
Elementos: 17 12
Escolha: 6
-> Destruído com sucesso!
Escolha: 7
-> Finalizando!
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista5/p2$
```