



Universidade Federal do Ceará
Disciplina: Programação – CK226 (2017/2)
Professor: Lincoln Souza Rocha
Email: lincoln@dc.ufc.br

Descrição da Atividade

- Tarefa de Programação 02 – Implementação de Fila usando Listas Encadeadas
- Entrega: 31/10/2018 às 23:59h via SIGAA
- Como: Enviar arquivo compactado contendo as implementações `Aluno.c` e `Fila.c`

Comandos para Compilação dos Módulos do Programa e Execução dos Testes:

```
$ gcc -c Aluno.c -o Aluno.o
$ gcc -c Fila.c -o Fila.o
$ gcc -c <prog-usa-fila-e-aluno>.c -o <prog-usa-fila-e-aluno>.o
$ gcc -o <prog-executavel>.bin Aluno.o Fila.o <prog-usa-fila-e-aluno>.o
$ ./<prog-executavel>.bin
```

As filas são estruturas de dados do tipo FIFO (*first-in first-out*), onde o primeiro elemento a ser inserido, será o primeiro a ser retirado, ou seja, adiciona-se itens no fim e remove-se do início. Nessa tarefa de programação você deverá implementar uma fila de alunos. Cada aluno é um tipo estruturado que possui os seguintes campos: matrícula (inteiro), nome (uma string de tamanho 50), curso (string de tamanho 30) e um ponteiro para o próximo Aluno. O tipo estruturado Fila deve possuir um campo do tipo ponteiro para Aluno e um campo inteiro tamanho. A seguir são dados os arquivos de cabeçalho de Aluno e Fila. Você deve implementar os arquivos `Aluno.c` e `Fila.c` e entregá-los via SIGG como discutido em sala de aula. A correção será feita através de testes automatizados, portanto, além de implementarem as funcionalidades (as funções descritas nos arquivos de cabeçalho) atentem para as restrições de entrada (tamanho, valoração, nulidade, etc.).

1) Descrição do arquivo de cabeçalho `Aluno.h`:

```
/* TAD: Aluno (matricula, nome, curso) */
typedef struct aluno Aluno;

/* Aloca e retorna um aluno com os dados passados por parâmetro */
Aluno *novo_a(int matricula, char *nome, char *curso);

/* Libera a memória de um aluno previamente criado */
void libera_a(Aluno *aluno);

/* Copia os valores de um aluno para as referências informadas */
void acessa_a(Aluno *aluno, int *matricula, char *nome, char *curso);

/* Atribui novos valores aos campos de um aluno */
void atribui_a(Aluno *aluno, int matricula, char *nome, char *curso);

/* Retorna o tamanho em bytes do TAD aluno */
int tamanho_a();
```

2) Descrição do arquivo de cabeçalho Fila.h:

```
#include "Aluno.h"

/* TAD: Fila (tamanho)*/
typedef struct fila Fila;

/* Aloca e retorna uma fila de tamanho informado */
Fila *novo_f(int tamanho);

/* Libera a memória de uma fila previamente criada */
void destroi_f(Fila *fila);

/* Adiciona um aluno na fila. Retorna 1 se der certo e 0 caso contrário */
int adiciona_f(Fila *fila, Aluno *aluno);

/* Remove um aluno na fila. Retorna 1 se der certo e 0 caso contrário */
int retira_f(Fila *fila);

/* Busca aluno pelo número de matrícula. Retorna o aluno se der certo e NULL
caso contrário */
Aluno *busca_f(Fila *fila, int matricula);

/* Retorna 1 se a fila estiver cheia e 0 caso contrário */
int cheia_f(Fila *fila);
```