

Programação

Laboratório 4 - Versão A

CAP Diogo Silva

Table of contents

Laboratório 4 - Carregar base de dados	1
Notas	1
Parte 1 - Adicionar um nó a lista ligada [30%]	1
Parte 2 - Escrever todos os elementos de lista ligada [20%]	2
Parte 3 - Integração [30%]	2
Parte 4 - Filtro [20%]	2
Código inicial	2
Ajuda	6

Laboratório 4 - Carregar base de dados

Notas

- Apenas é permitido o uso do computador para
 - aceder ao replit.com (para resolução do laboratório)
 - aceder ao conteúdo das aulas e exercícios passados
 - aceder aos livros da bibliografia
- A resolução dos laboratório é individual.
- No final, submeter a solução no replit.

Parte 1 - Adicionar um nó a lista ligada [30%]

- Implementa a função `addEstudante` que recebe uma lista ligada de `Estudante` e um `Estudante` para adicionar.
- A função cria um novo nó na lista com os dados do `Estudante` recebido.

- Nota: se não conseguires implementar esta função, tenta implementar a mesma funcionalidade diretamente na `main`.

Parte 2 - Escrever todos os elementos de lista ligada [20%]

- Implementa a função `printEstudantes` que recebe uma lista ligada de `Estudante` e escreve na consola todos os elementos da lista.
- Usa a função `printEstudante` para escrever na consola cada um dos elementos da lista.

Parte 3 - Integração [30%]

- Escreve a função `main` de modo a que:
 - o programa receba um vetor de `n` estudantes com dados aleatórios
 - * usa a função `createEstudantes` para este efeito
 - crie uma lista ligada de `Estudante`
 - adicione cada um dos estudantes do vetor à lista ligada (`addEstudante`)
 - liberte a memória do vetor
 - escreva na consola todos os elementos da lista ligada (`printEstudantes`)

Parte 4 - Filtro [20%]

- Modifica a função `main` de modo a que, no final:
 - elimina da lista ligada todos os estudantes com média inferior a 17
 - * para cada estudante removido, escrever na consola `removido estudante <nome> com media <media_final>`
 - escreve na consola os alunos do quadro de honra (média ≥ 17) (`printEstudantes`)

Código inicial

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* * * * * *
 * ESTRUTURAS *
 * * * * * */

typedef struct {
```

```

    int id;
    char nome[50];
    float media_final;
} Estudante;

// linked list node of type Estudante
typedef struct node {
    Estudante estudante;
    struct node *next;
} Node;

// function to print a Estudante pointer
void printEstudante(Estudante *estudante) {
    printf("ID: %d\n", estudante->id);
    printf("Nome: %s\n", estudante->nome);
    printf("Media Final: %.2f\n", estudante->media_final);
}

/* * * * * *
 * MOCK DATA *
 * * * * * */

// array of 20 portuguese first names
char *nomes_proprios[] = {
    "Maria",
    "Joao",
    "Ana",
    "Pedro",
    "Carla",
    "Jose",
    "Margarida",
    "Miguel",
    "Sofia",
    "Ricardo",
    "Ines",
    "Paulo",

```

```

        "Beatriz",
        "Tiago",
        "Catarina",
        "Andre",
        "Filipa",
        "Bruno",
        "Diana",
        "Rui"
    };

    // array of 20 portuguese last names
    char *apelidos[] = {
        "Silva",
        "Santos",
        "Rodrigues",
        "Costa",
        "Oliveira",
        "Martins",
        "Ferreira",
        "Pereira",
        "Carvalho",
        "Marques",
        "Gomes",
        "Sousa",
        "Fernandes",
        "Goncalves",
        "Ribeiro",
        "Leal",
        "Faria",
        "Henriques",
        "Neves",
        "Oliveira"
    };

    Estudante *createEstudantes(unsigned int n) {
        Estudante *estudantes = malloc(n * sizeof(Estudante));
        for (int i = 0; i < n; i++) {

```

```

        estudantes[i].id = 1400 + i;
        int rand_nome = rand() % 20;
        int rand_apelido = rand() % 20;
        sprintf(estudantes[i].nome, "%s %s", nomes_proprios[rand_nome], apelidos[rand_apelido]);
        estudantes[i].media_final = 5 + rand() % 16;
    }
    return estudantes;
}

/* * * * * *
 * LISTA LIGADA *
 * * * * * */

// function to add estudante to list
void addEstudante(Node **head, Estudante estudante) {
}

// function to print a linked list of Estudante
void printEstudantes(Node *head) {
}

int main(){
    // podes alterar este valor para testar com arrays de tamanhos diferentes
    int n_estudantes = 10;

    // receber array de Estudantes aleatórios, com a função createEstudantes

    // criar lista ligada

    // adicionar todos os Estudante do array à lista ligada

    // libertar array de Estudante da memória

    // escrever na consola todos os Estudante da lista ligada

```

```

// eliminar da lista ligada todos os Estudante com media_final < 17
//     para cada estudante removido, escrever na consola "removido estudante <nome> co

// imprimir quadro de honra (todos os alunos com média final >= 17, i.e. os alunos que
printf("\nQuadro de Honra:\n");
// ...

}

```

Ajuda

Se ajudar, podes criar outra estrutura de dados para ter a cabeça da lista ligada e modificar as funções para receber uma lista desse tipo, e.g.

```

typedef struct {
    Node * head;
} Lista;

void addEstudante(Lista * lista, Estudante estudante) {

}

// function to print a linked list of Estudante
void printEstudantes(Lista * lista) {
}

```