

<b>Estudante:</b>	<b>Matrícula:</b>
-------------------	-------------------

**Instruções:**

- Este exame possui 2 página(s). Verifique se sua cópia do exame está completa.
- Esta prova é sem consulta:** você não tem permissão para consultar o livro texto, suas notas de aula, a Internet, seus colegas ou quaisquer outras pessoas ou fontes para concluir o exame.
- Se você acredita que alguma pergunta esteja subespecificada, anote as suposições que você teve que fazer para chegar a sua resposta e justifique-as como parte de sua resposta à pergunta.
- Lembre-se de indentar o código** de maneira apropriada e atente-se a organização, clareza e legibilidade do código!

---

Utilize os espaços a seguir, referentes aos arquivo de cabeçalho `curso.h` e `aluno.h`, para a definição das estruturas solicitadas na primeira questão (se atente aos nomes dos campos!).

**curso.h**

```
1 #ifndef CURSO_H_
2 #define CURSO_H_
3 typedef struct Curso {
4
5
6
7
8
9 } Curso;
10 #endif // CURSO_H_
```

**curso.h**

```
1 #ifndef ALUNO_H_
2 #define ALUNO_H_
3
4 #include <stdbool.h>
5 #include "curso.h"
6
7 typedef struct Aluno {
8
9
10
11
12
13 } Aluno;
14
15 typedef struct VetorAlunos {
16
17
18
19
20
21 } VetorAlunos;
22
23 bool salvarAlunos(VetorAlunos v, char *nome_arquivo);
24 #endif // ALUNO_H_
```

1. (4 pontos) Complete as estruturas `Curso`, `Aluno` e `VetorAlunos`, presentes no arquivos de cabeçalho `curso.h` e `aluno.h`, de acordo com as seguintes especificações (dica: lembre-se de considerar o `\0` nas strings):
  - (a) (1 ponto) `Curso`: `id` (inteiro não negativo), `nome` (string, tamanho variável, pode conter espaços) e `departamento` (string, sigla de 3 caracteres)
  - (b) (1,5 pontos) `Aluno`: `matricula` (string, 6 dígitos), `nome` (string, tamanho variável, pode conter espaços) e `curso` (ponteiro para a estrutura `Curso`)
  - (c) (1,5 pontos) `VetorAlunos`: `alunos` (arranjo de `Alunos`, tamanho variável), `capacidade` (inteiro não negativo) e `n` (inteiro não negativo)
2. (a) (3 pontos) Complete a função `f` a seguir.

**aluno.c**

```

1 VetorAlunos f(VetorAlunos v, Curso curso) {
2     VetorAlunos r;
3
4     r.capacidade = 10;
5     r.n = 0;
6
7     // Aloque espaco de acordo com a capacidade inicial:
8     r.alunos = _____;
9
10    for (int i = 0; i < v.n; i++) {
11        if (r.n >= r.capacidade) {
12            // Dobre a capacidade:
13            r.capacidade *= _____;
14
15            // Realoque o espaco para comportar novos alunos:
16            r.alunos = _____;
17        }
18
19        if (v.alunos[i].curso->id == curso.id) {
20            r.alunos[r.n] = v.alunos[i];
21            r.n++;
22        }
23    }
24
25    return r;
26 }

```

- (b) (6 pontos) Descreva o que faz a função `f` apresentada acima. Como você renomearia a função `f`? Mostre a execução e o resultado da função, caso ela tenha sido chamada com o segundo parâmetro sendo o curso de Ciência da Computação (ID 1) e o primeiro parâmetro sendo um vetor com os seguintes alunos:
  - i. Matrícula: 111111, Nome: Chico, Curso: Ciência da Computação (DCC, ID 1)
  - ii. Matrícula: 222222, Nome: Dara, Curso: Enfermagem (ENA, ID 2)
  - iii. Matrícula: 333333, Nome: Mário, Curso: Ciência da Computação (DCC, ID 1)
- (c) (6 pontos) Implemente a função `f` acima de forma **RECURSIVA**. Isto é, sua função **NÃO pode conter laços de repetição (for, while, do-while)**. Sua função deve seguir o mesmo protótipo da função `f`: o retorno deve ser uma estrutura `VetorAlunos`, e deve possuir dois parâmetros dos tipos: `VetorAlunos` e `Curso`. Se desejar, você pode criar funções auxiliares, mas estas também **NÃO podem conter laços de repetição**.
3. (6 pontos) Implemente uma função `salvarAlunos`, que recebe como parâmetro um vetor de alunos e o nome do arquivo (verifique o arquivo de cabeçalho `aluno.h`). Você deve criar o arquivo caso ele não exista ou sobrescrever o conteúdo do arquivo se ele já existir. Você deve salvar a matrícula e o nome de cada aluno, além do ID do curso (um aluno por linha). Você deve retornar falso caso não tenha sido possível abrir o arquivo ou verdadeiro caso contrário. O arquivo final deve obedecer o seguinte formato:

```

1 Matricula,Nome,Curso
2 111111,Chico,1
3 222222,Dara,2

```