

Estruturas de Dados

Arquivos – Exercícios

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Considere um arquivo texto com as notas dos alunos de uma disciplina. Os dados de cada aluno são armazenados em duas linhas do arquivo: uma com o seu nome (cadeia com até 80 caracteres), e outra com suas três notas (P1, P2 e P3). Considere ainda que não existem linhas em branco no arquivo. Um exemplo desse formato é mostrado abaixo.

```
João da Silva
2.0 4.3 6.5
Manuel Santos
7.0 8.2 8.6
Fulana de Tal
6.0 7.5 7.8
```

Escreva um programa completo que leia o arquivo “entrada.txt”, que contém as notas de uma disciplina no formato descrito acima, e escreva no arquivo texto “saida.txt” o nome de cada aluno com sua respectiva média $((P1+P2+P3)/3)$ e um caractere indicando se o aluno foi aprovado ou reprovado na disciplina (use ‘A’ para indicar aprovação e ‘R’ para indicar reprovação). Considere que um aluno está aprovado se tiver média maior ou igual a 5.0. Para o exemplo acima, o programa deve criar um arquivo “saida.txt” com o seguinte conteúdo:

```
João da Silva 4.3 R
Manuel Santos 7.9 A
Fulana de Tal 7.1 A
```

Observe que as médias são escritas no arquivo “saida.txt” com apenas uma casa decimal. Se não for possível abrir um dos arquivos, o programa deve imprimir a mensagem “ERRO” e terminar sua execução

```

#include <stdio.h>
#include <string.h>
int main( )
{
    char nome[81];
    float p1,p2,p3,media;

    FILE *fp_in=fopen("entrada.txt","rt");
    FILE *fp_out=fopen("saida.txt","wt");
    if (fp_in==NULL || fp_out==NULL) {
        printf("ERRO\n"); return 0; }

    while(fscanf(fp_in,"%80[^\n] %f %f %f ",nome,&p1,&p2,&p3)==4)
    {
        media=(p1+p2+p3)/3;
        if (media>=5)
            fprintf(fp_out,"%s %.1f A\n",nome,media);
        else
            fprintf(fp_out,"%s %.1f R\n",nome,media);
    }

    fclose(fp_in); fclose(fp_out);
    return 1;
}

```

```

#include <stdio.h>
#include <string.h>
int main( ) {
char nome[81];
float p1,p2,p3,media;

FILE *fp_in=fopen("entrada.txt","rt");
FILE *fp_out=fopen("saida.txt","wt");
if (fp_in==NULL || fp_out==NULL) {
printf("ERRO\n"); return 0; }

while(fgets(nome,80,fp_in)!=NULL) {
int n=strlen(nome); nome[n]='\0';
fscanf(fp_in, "%f %f %f ",&p1,&p2,&p3);
media=(p1+p2+p3)/3;
if (media>=5)
fprintf(fp_out,"%s %.1f A\n",nome,media);
else
fprintf(fp_out,"%s %.1f R\n",nome,media);
}
fclose(fp_in); fclose(fp_out);
return 1;

```

Questão 2

Considere um tipo que representa um funcionário de uma empresa, definido pela estrutura a seguir:

```
struct funcionario {
    char nome[81]; /* nome do funcionário */
    float valor_hora; /* valor da hora de trabalho em Reais */
    int horas_mes; /* horas trabalhadas em um mês */
};
typedef struct funcionario Funcionario;
```

Escreva uma função em C que preencha um vetor de ponteiros para `Funcionario` com os dados lidos de um arquivo texto. Essa função deve receber como parâmetros o vetor de ponteiros para `Funcionario` (representado pelo seu comprimento e pelo endereço de seu primeiro elemento) e o nome do arquivo de entrada. Nesse arquivo de entrada, os dados de cada funcionário são armazenados em duas linhas: uma com o seu nome (cadeia com até 80 caracteres), e outra com o valor de sua hora de trabalho e com o número de horas trabalhadas em um mês (nessa ordem). Um exemplo desse formato é mostrado abaixo.

```
João da Silva
15.0 160
Manuel Santos
15.0 80
Fulana de Tal
23.5 40
```

A função implementada deve ter o seguinte protótipo:

```
void carrega (int n, Funcionario** vet, char* arquivo);
```

Considere que: não existem linhas em branco no arquivo; o parâmetro `vet` já vem com todas as suas posições inicializadas com `NULL`; o comprimento do vetor (parâmetro `n`) é no mínimo igual à quantidade de registros de funcionários no arquivo de entrada; se não for possível abrir o arquivo, a função deve imprimir a mensagem “ERRO” e terminar a execução do programa.

Questão 2

```
void carrega (int n, Funcionario** vet, char* arquivo) {
    char nome[81];
    float v;
    int h, i = 0;
    FILE* f = fopen (arquivo, "rt");
    if (f == NULL) {
        printf("ERRO\n");
        exit(1);
    }
    while (fscanf(f, " %80[^\n] %f %d", nome, &v, &h) == 3) {
        vet[i] = (Funcionario*) malloc(sizeof(Funcionario));
        strcpy(vet[i]->nome, nome);
        vet[i]->valor_hora = v;
        vet[i]->horas_mes = h;
        i++;
    }
    fclose(f);
}
```