

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO - DEPARTAMENTO DE INFORMÁTICA
Lista 5 - PDII e PROGII

1. Faça um programa para ler os nomes e os coeficientes de rendimento de cada mês de cada aluno de uma turma. Calcule o coeficiente médio de cada aluno e imprima na tela os meses em que o aluno estava acima de sua própria média. Use struct e malloc.
2. Faça o mesmo programa aplicado a várias turmas com números de alunos diferentes. Além disso, imprima o nome do aluno que obteve a maior média de coeficientes entre todas as turmas.
3. Considere que exista um arquivo chamado *voo.dat* que armazena diversos vôos entre várias cidades. Cada vôo contém as seguintes informações: número do vôo, origem, destino, valor da passagem e empresa aérea. Faça um programa para informar ao interessado na compra de uma passagem se existem vôos entre as cidades fornecidas pelo teclado. A saída do seu programa, quando não existir o vôo desejado será **NÃO EXISTEM VÔOS**. Caso existam, imprimir o número do vôo, valor da passagem e a empresa. Pode haver mais de um vôo entre as cidades desejadas. Use struct em seu programa. Para comparar strings use a função `strcmp(st1,st2)` que tem os seguintes valores de retorno: (a) valor 0 se o `st1=st2`; (b) valor -1 se o `st1<st2` considerando ordem alfabética; (c) valor 1 se o `st1>st2` considerando ordem alfabética.

Exemplo:

`strcmp(cris,cris) ← 0; strcmp(cris,zezinho) ← -1; strcmp(zezinho,cris) ← 1.`

4. Considere um arquivo chamado 'paciente.dat' que armazena os dados dos pacientes de um endocrinologista. O arquivo é organizado onde cada paciente tem seguintes dados:

nome do paciente
idade peso altura

Faça um programa, usando struct, que leia os dados e guarde em um arquivo chamado 'fofinhos.dat' os nomes e as idades dos pacientes que estão acima do peso ideal, calculado por uma função, fornecido pela relação

$$PesoIdeal = (altura - 1) \times 100$$

5. Existem 5 turmas `t1.txt`, ... ,`t5.txt` que armazenam número de alunos, nomes e notas finais de cada aluno. Faça um programa para acessar os arquivos e verificar, a cada turma, os alunos com nota final acima (ou igual) de 7 e gravar os respectivos nomes em outro arquivo `aprovados.txt`. Este arquivo `aprovados.txt` deve armazenar todos os alunos aprovados, independentemente da turma de origem. Use struct.