

# Prova 2

## Programação e Desenvolvimento de Software I

**Professor:** Pedro O.S. Vaz de Melo

Nome: \_\_\_\_\_

escrevendo o meu nome eu juro que seguirei o código de honra.

### Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca **prova1.h**), função **main**, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- As funções implementadas no módulo **prova2.h** podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo **prova2.h**, considere que ela está implementada de forma correta.

Os exercícios a seguir servirão de base para um programa de análise de contusões de atletas. Para eles, faça uso do código abaixo, que está localizado no **prova2.h**. Nos exercícios, você pode tanto usar as definições (ex: **NUM\_DIAS**) ou os seus valores diretamente (ex: 7).

```
#define NUM_DIAS 7
#define NUM_ATLETAS 10
#define NUM_EQUIPES 30

typedef struct Atleta {
    float carga[NUM_DIAS];
    int contundiu;
} Atleta;

float randf(float min, float max) {
    return min + (max-min)*(float)rand()/RAND_MAX;
}
```

1. (4 points) Escreva uma função de nome **preencheCarga** que recebe um vetor de pontos flutuantes **x** como parâmetro, o seu tamanho **n**, e o transforma em uma série de carga de treino diária de um atleta. A posição 0 do vetor refere-se à carga de treino, em horas, do primeiro dia de treino, a posição 1, do segundo dia, e assim por diante. A carga diária de cada atleta é um número de horas aleatório (que pode ser fracionado) entre 4 e 12, exceto nos dias em que no dia anterior o atleta tenha treinado mais de 8 horas. Nesse caso, a carga diária é um número de horas aleatório entre 0 e 4 (que também pode ser fracionado). Protótipo:

```
void preencheCarga(float x[], int n);
```

2. (4 points) Escreva uma função que recebe um vetor de pontos flutuantes **x** como parâmetro, o seu tamanho **n**, e retorna a probabilidade de contusão de um atleta dada a sua carga de treino semanal. A probabilidade é simplesmente  $\frac{x_{max}}{x_{max} + \bar{x}}$ , em que  $x_{max}$  é o maior valor em **x** e  $\bar{x}$  é a média aritmética de **x**. Protótipo:

```
float probContusao(float x[], int n);
```

3. (5 points) Escreva uma função `initAtleta` que recebe um `Atleta` como parâmetro **por referência** e preenche os seus campos. Use a função `preencheCarga` para preencher a carga de treinos do atleta. O campo `contundiu` indica se o atleta contundiu (valor 1) ou não (valor 0) depois dos treinos. Para preencher esse campo, gere um valor aleatório ponto flutuante entre 0 e 1 e, caso o valor seja menor que a sua probabilidade de contusão (use a função `probContusao`), indique que o atleta contundiu. Caso contrário, indique que ele não contundiu.

4. (4 points) Escreva uma função de nome `fracContundidos` que retorna a fração de atletas contundidos em um vetor de atletas. A sua função recebe um vetor de atletas `a` e o seu número de elementos `n`. Protótipo:

```
float fracContundidos(Atleta a[], int n);
```

5. (4 points) Escreva uma função de nome `imprimeTime` que recebe um vetor de atletas como parâmetro e imprime as suas informações. Formato de impressão de um atleta fictício, localizado na posição 4 do vetor:

```
4: 10.91 0.92 6.00 8.34 3.94 4.43 4.65 (contundiu: 1)
```

Imprima um atleta por linha. Protótipo:

```
void imprimeTime(Atleta a[], int n);
```

6. (6 points)

Complete o programa abaixo. Este programa cria equipes de atletas usando a função `initAtleta`. As equipes estão nas linhas da matriz `times`, sendo que cada linha é o vetor de atletas que faz parte da equipe. Depois de preencher as equipes, você deve imprimir a menor fração de atletas contundidos entre as equipes.

```
#include _____ (0.5)
```

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

```
void main() {
```

```
    Atleta times[NUM_EQUIPES][NUM_ATLETAS];
```

```
    int i,j;
```

```
    float frac, minFrac = _____; (0.5)
```

```
    for(_____) { (1)
```

```
        for(_____) (1)
```

```
            _____
            imprimeTime(times[i], NUM_ATLETAS);
```

```
            frac = _____; (1)
```

```
            if(_____) (1)
```

```
                _____; (1)
```

```
        }
```

```
        printf("Menor fracao de contundidos: %.2f", minFrac);
```

```
    }
```