



## Prova de Seleção 2012/2

23 de Julho de 2012

**Nome do Candidato:** \_\_\_\_\_

**Aviso:** programas devem ser escritos em C ou C++ **Valor:** 10 pontos. **Nota:** \_\_\_\_\_

1. (2 pontos) Crie uma estrutura (tipo) para armazenar um ponto bidimensional. Crie uma estrutura para armazenar um círculo: essa estrutura armazenará o seu centro (tipo ponto) e o raio. Faça um programa que leia os dados de um círculo e em seguida leia os dados de um ponto. Ao final o programa deve responder se o ponto está contido no círculo (criar uma função distância que receba como dados de entrada os dois pontos).
2. (1 ponto) Um dado é lançado  $n$  vezes e a cada lançamento é anotado o resultado. Faça um algoritmo que tendo como entrada o resultado de cada um dos  $n$  lançamentos, encontra a média dos lançamentos e conta quantos lançamentos estão acima da média dos lançamentos.

Exemplo:  $n = 5$

Lançamento	Resultado
1	6
2	3
3	1
4	5
5	5

*Média dos Lançamentos* =  $(6 + 3 + 1 + 5 + 5) / 5 = 4$

*Número de Lançamentos que estão acima da média* = 3  
(lançamentos 1, 4 e 5)

3. (2 pontos) Criar a função que responda se uma cadeia está contida dentro de outra. Por exemplo, a cadeia "aluno" está contida dentro da cadeia "aluno no".
4. (2 pontos) Escreva uma função que receba duas strings,  $s1$  e  $s2$  e retorne a maior substring de  $s1$  que aparece em  $s2$ . Por exemplo:

$s1$ : abacaxi  
 $s2$ : bacalhau

Maior substring: baca. Dica: utilize a função implementada na questão anterior.

5. (2 pontos) Suponha que lhe seja pedido determinar se um dado número pertence a um conjunto ordenado de números. A ordenação permite que não sejam necessárias  $n$  comparações, onde  $n$  é o tamanho do vetor de entrada. A resolução

deste problema é feita por um dos algoritmos mais importantes da computação. Implemente uma versão sequencial e diga qual a complexidade de tempo da sua solução.

6. (1 ponto) Escreva uma função para computar uma função de escrutínio (hash): dado um número qualquer de entradas diferentes no vetor *coordinates* a função abaixo deve retornar um valor entre  $\{0, \dots, m-1\}$ .

```
int compute_hash_code(int coordinates[3], int m )
{

int h;

/* escreva abaixo o código para calcular h em função de
coordinates[3], considerando que h deve, dentro do possível,
receber diferentes valores para diferentes entradas, mas
sempre  $0 \leq h < m$  */

}
```