

## Prova 2

- Escreva as respostas com caneta.
- Você pode criar funções auxiliares se desejar, mas não considere a pré-existência de funções.
- Não é permitido usar strings de C++ ou funções prontas da biblioteca strings.h de C.
- **Não utilize funções de qualquer biblioteca padrão além de printf().**
- Atribua de 1,0 à 3,0 pontos para cada questão, não ultrapassando o total de 10,0 pontos.

- 1) ( ) Escreva uma função que recebe uma string representando um número binário. A função deve converter o número para decimal (inteiro) e devolvê-lo. Dica:  $1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18$

```
int binaryToDecimal(char bin[])
```

```
Ex: int num = binaryToDecimal("10010"); // num = 18
```

- 2) ( ) Escreva uma função que recebe dois arrays embaralhados informa se eles são similares: possuem o mesmo tamanho e os mesmos elementos (em qualquer ordem). A função deve devolver true (1) ou false (0). OBS: considere que não há elementos repetidos em um mesmo vetor.

```
int areSimilar(int v1[], int n1, int v2[], int n2)
```

```
Ex: {1,2,5,6,7} é similar a {5,2,1,7,6}
```

- 3) ( ) Escreva uma função que recebe um array e remove todos os elementos repetidos. O array contém somente inteiros positivos ( $x > 0$ ). As posições livres (ao final) deverão ser preenchidas com 0 (zero).

```
void removeClones(int v[], int n)
```

```
Ex: {1,2,3,2,5,2,1,3,4} após a função deve ficar: {1,2,3,5,4,0,0,0,0}
```

- 4) ( ) Escreva uma função que verifica se uma string forma um palíndromo: expressões que podem ser lidas da esquerda para a direita ou da direita para a esquerda tendo sentidos mantidos, mas desconsiderando espaços. A função deve retornar true (1) ou false (0).

```
int isPalindrome(char str[])
```

Exemplos de strings palíndromos:

“ame o poema”, “a sacada da casa”, “luz azul”, “arara”, “ana”