

Prova 2

- Escreva as respostas com caneta.
- Você pode criar funções auxiliares, se desejar. Contudo, não considere a pré-existência de funções prontas e não utilize funções da biblioteca além de `printf()`, `scanf()`, `pow()` e `sqrt()`
- Atribua de 1,0 à 3,0 pontos para cada questão, não ultrapassando o total de 10,0 pontos.

- 1) () Escreva uma função que faz a leitura de **n** números inteiros positivos ($x \geq 0$) e os imprime sem repetições.

void readUniqueNumbers(int n)

Ex: $n=9$, digitados: 1 4 1 8 3 7 1 8 4 \Rightarrow saída: 1 4 8 3 7

- 2) Escreva uma função que recebe uma string e inverte as letras maiúsculas/minúsculas. O texto pode conter outros caracteres além de letras, que devem permanecer inalterados.

Dica: na tabela ASCII, 'a'= 97 'A'=65.

void stringSwitch(char str[])

Ex: `char s[] = "All your 2 BASES are Belong to US!";`
`stringSwitch(s);`
`printf("%s", s);` // saída: aLL YOUR 2 bases ARE bELONG TO us!

- 3) Escreva uma função que mescla (alterna) os elementos de dois vetores **v1** e **v2**, colocando-os em **v3**. Considere que **v3** tem tamanho **n1 + n2**. A mesclagem inicia por **V1**.

void simpleMerge(int v1[], int n1, int v2[], int n2, int v3[])

Exemplo:	Entrada	$v1 = \{1, 2, 3\}$
		$v2 = \{10, 20, 30, 40, 50\}$
	Saída	$v3 = \{1, 10, 2, 20, 3, 30, 40, 50\}$

- 4) Escreva uma função que converte um número inteiro positivo (parâmetro **number**) para uma string contendo a representação binária do mesmo (parâmetro **binary**). Dica: Utilize módulo (%) e divisão (/) para auxiliá-lo no processo de obtenção do número binário. Assuma que a string tem espaço suficiente.

void binaryString(int number, char binary[])

Ex: `char num[10];`
`binaryString(128, num);`
`printf("%s", num);` // saída: "10000000"