



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL
TÉCNICAS DE PROGRAMAÇÃO – PROF. FISCHER

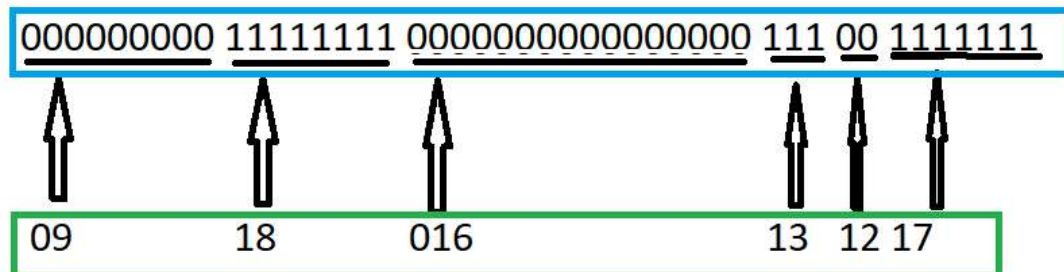
Lista 3

Faça uma função para cada um dos itens abaixo. Você deverá testar cada função com dados que exercitem a lógica de cada uma das funções abaixo.

- 1) **Pesquisa:** Sua função deverá receber um vetor e retornar se um dado elemento está dentro do vetor informado. Caso o elemento esteja no vetor retornar a posição do elemento no vetor, caso não esteja retornar -100;
- 2) **Pesquisa por índice:** Sua função deverá receber um vetor e retornar o elemento do vetor por meio de um índice informado. Seu programa deverá tratar os casos em que o índice não existe no vetor.
- 3) **Separação de valores positivos e valores negativos:** Sua função deverá receber um vetor e deverá colocar nesse vetor os números negativos nas primeiras posições e os números positivos na últimas posições;
- 4) **Somatório:** Sua função deverá receber um vetor e retorna a soma de todos elementos do vetor;
- 5) **Junção de vetor:** Sua função deverá receber dois vetores de qualquer tamanho e retornar apenas um vetor que contenha os dois vetores. O primeiro vetor deverá ficar nas primeiras posições e o segundo vetor nas últimas posições.
- 6) **Inversão de ordem:** Sua função deverá retornar um vetor de ordem inversa. Desta forma, o primeiro elemento passa ser o último elemento do vetor, o segundo elemento passa ser o penúltimo, assim, sucessivamente;
- 7) **Inversão de vetores:** Sua função deverá receber dois vetores. Sendo que os dados do primeiro vetor deve ser colocado no segundo vetor e os dados segundo vetor devem ser colocados no primeiro vetor.
- 8) **Priorização:** Sua função deverá receber um vetor e pegar os elementos do vetor e colocar em um segundo vetor. Os novos elementos devem ser colocados no segundo vetor apenas quando eles contribuam para a média dos segundo vetor aumentar.
- 10) **Compactação** Sua função deverá receber um vetor composto com uma sequência de zeros e uns e gerar um novo vetor compactado.

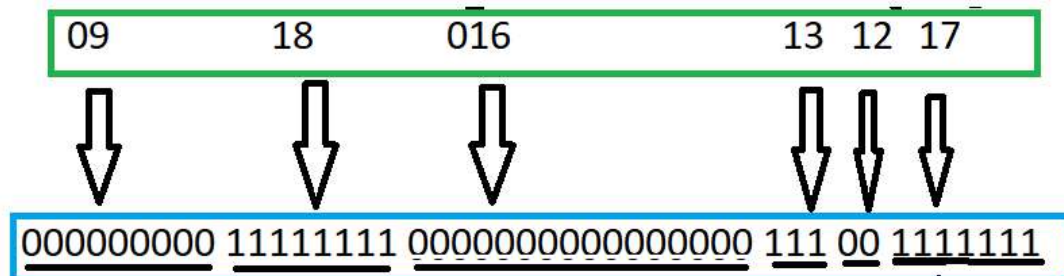
A quantidade dos elementos repetidos devem ser contados e o número de ocorrências deve ser colocado logo após o elemento.

Por exemplo: O vetor de entrada está indicado em azul “apenas zeros e uns” a saída está destacado em verde.



11) Descompactação: Sua função deverá receber um vetor e gerar um vetor descompactado. O número de repetições devem ser colocadas no novo vetor conforme o número indicado.

Por exemplo: O vetor de entrada está indicado em verde e a saída está destacada em azul.



12) Criptografia: Sua função deverá criptografar os caracteres do vetor de strings segundo a lista abaixo. Desta forma todas as palavras no vetor deverão ser criptografadas.

- a = %
- b = *
- c= (
- d= -
- e=+
- f=@
- g=#
- h=1

- i= 2
- j=3
- l=4
- m=5
- n=6
- o=7
- p=8
- q=9
- r={
- s=}
- t=!
- u=&
- v=\$
- x=?
- z=:

13) **Descritografia.** Sua função deverá descriptografar os caracteres do vetor de strings segundo a lista apresentada na questão anterior.

14) **Contagem de caracteres:** Sua função deverá contar os caracteres que aparecem no vetor. Os caracteres podem ser qualquer letra ou número do alfabeto.

Exemplo:

vetor de entrada {a,b,c,3,3,5,a,c}

resultado: a=2 - b=1 - c=2 - 3=2 - 5=1

15) **Complementação:** Sua função deverá receber dois vetores de números inteiros positivos e negativos. Ela deverá produzir um terceiro vetor em que cada índice desse vetor tenha um número que somado com elementos do mesmo índice dos vetores anteriores tenha a soma sempre igual a 10.

Exemplo:

vetor a) {1, 2, 4, 5, 6}

vetor b) {5, 1, 9, 4, 2}

vetor resultante {4, 7, -3, 1, 2}

16) **Polarização:** Sua função deverá receber um vetor com números aleatórios. Os números ímpares devem ficar nas posições iniciais do vetor e os números pares nas posições finais.

17) **Explosão:** Sua função deverá receber um vetor e gerar um segundo vetor com os dados acrescidos. O número de cada posição do vetor de entrada corresponde ao número de elementos que devem ser gerados.

Exemplo:

Vetor de entrada: {2,3,2,4,3}

vetor de saída: {2,2,3,3,3,2,2,4,4,4,3,3,3}

18) **Organização:** Sua função deverá receber um vetor e organizar os elementos a fim de que os elementos iguais fiquem próximos.

Exemplo:

vetor de entrada: {1,2,3,1,2,3,4,5}

organização: {1,1,2,2,3,4,5}

19) **Agrupamento:** Sua função deverá receber o número de agrupamento desejado e os elementos devem ser agrupado conforme o número de agrupamento informado;

Exemplo com dois grupos: Os elementos do vetor deverão ser divididos em dois vetores.

20) **Diferença-A:** são informados dois vetores (A e B) e o programa deverá mostrar todos elementos que estão no vetor A e não estão no vetor B;