

Atividade Sábado

DESCRIÇÃO

Cada vez mais os processos seletivos estão utilizando resolução de problemas para analisar o nível de compreensão dos conceitos de programação do candidato. Para simular esse processo será exigido a entrega de no **mínimo quatro** dos algoritmos descritos a seguir.

QUESTÕES

QUESTÃO 1 - Crie um algoritmo que recebe uma data no formato “dd/mm/aaaa” e escreva a data por extenso. Então, a entrada será uma data no formato de texto usando o separador de barra (/) entre os campos da data. Já a saída deve possuir o formato por extenso dada a entrada, por exemplo, na data 01/01/1970 deve ser retornado 01 de janeiro de 1970.

Para analisar mais exemplos veja a *Tabela 1.1*.

Tabela 1.1:

Entrada	Saída
29/10/1969	29 de outubro de 1969
01/01/1970	01 de janeiro de 1970
15/09/1975	15 de setembro de 1975
01/04/1976	01 de abril de 1976
01-04-1976	Invalid data input

1976	Invalid data input
aaa	Invalid data input

QUESTÃO 2 - Ao receber um texto transforme seu conteúdo para que as letras sejam exibidas em maiúsculas e separadas por hífen (-), contudo os espaços em branco devem ser desconsiderados. Veja alguns exemplos deste processamento na *Tabela 2.1*:

Tabela 2.1:

Entrada	Saída
programador	P-R-O-G-R-A-M-A-D-O-R
o dia está chuvoso	O-D-I-A-E-S-T-Á-C-H-U-V-O-S-O

QUESTÃO 3 - Crie um programa que ao receber um conjunto de meias identificadas por números retorne quantos pares do mesmo tipo é possível montar (Fonte: Sock merchant - Hackerrank).

Por exemplo, ao receber o conjunto de meias {1, 2, 1, 2, 1, 3, 2} veja que do total existem 3 tipos de meias {1, 2, 3}. Ao se agrupar em pares é possível formar 1 par do tipo de meia 1, e 1 do 2, o que totaliza 2 pares. Conforme é possível ver nas meias destacadas {1, 2, 1, 2, 1, 3, 2}.

Do mesmo modo, ao receber o conjunto de meias {10, 20, 20, 10, 10, 30, 50, 10, 20} irão existir esses 4 tipos de meias {10, 20, 30, 50} que ao se agrupar em pares é possível formar 2 pares do tipo de meia 10, e 1 do 20, o que totalizam 3 pares. Conforme é possível ver nas meias destacadas {10, 20, 20, 10, 10, 30, 50, 10, 20}.

Para analisar mais exemplos veja a *Tabela 3.1*.

Tabela 3.1

Entrada	Saída
1, 2, 3	0
1, 1, 1, 2	1
1, 2, 1, 2, 1, 3, 2	2
10, 20, 20, 10, 10, 30, 50, 10, 20	3

QUESTÃO 4 - Crie um algoritmo que rotacione os elementos de um array a em d saltos (Fonte: Arrays: Left Rotation - Hackerrank). Por exemplo, caso receba um array com os valores 1, 2, 3, 4, 5 e precisasse realizar 4 rotacionamentos o processamento seria:

Para analisar mais exemplos veja a *Tabela 4.1*.

Tabela 4.1

Entrada	Saída
1, 2, 3, 4, 5, 1	2, 3, 4, 5, 1
1, 2, 3, 4, 5, 4	5, 1, 2, 3, 4
41, 73, 89, 7, 10, 1, 59, 58, 84, 77, 77, 97, 58, 1, 86, 58, 26, 10, 86, 51, 10	77, 97, 58, 1, 86, 58, 26, 10, 86, 51, 41, 73, 89, 7, 10, 1, 59, 58, 84, 77
33, 47, 70, 37, 8, 53, 13, 93, 71, 72, 51, 100, 60, 87, 97, 13	87, 97, 33, 47, 70, 37, 8, 53, 13, 93, 71, 72, 51, 100, 60

QUESTÃO 5 - Receba um array e retornar a proporção de elementos menor, igual e maior que zero (Fonte: Plus Minus - Hackerrank).

Por exemplo, considerando inicialmente os valores `-4, 3, -9, 0, 4, 1` os elementos menor, igual e maior que zero são respectivamente `-4, -9, 0` e `3, 4, 1`. Na proporção representam os valores $\frac{2}{6}$, $\frac{1}{6}$, $\frac{3}{6}$ que resultam, na precisão de 6, em `0.333333`, `0.166667`, `0.500000`.

Para analisar mais exemplos veja a *Tabela 5.1*.

Tabela 5.1

Entrada	Saída
<code>1, 1, 1</code>	<code>0.000000, 0.000000, 1.000000</code>
<code>0, 0, 1, 1</code>	<code>0.000000, 0.500000, 0.500000</code>
<code>-4, 3, -9, 0, 4, 1</code>	<code>0.333333, 0.166667, 0.500000</code>
<code>1, 2, 3, -1, -2, -3, 0, 0</code>	<code>0.375000, 0.250000, 0.375000</code>

QUESTÃO 6 - Receba um array de tamanho 5 e retorna o valor mínimo e máximo da soma de 4 elementos do array (Fonte: Mini-Max Sum - Hackerrank).

Por exemplo, considerando inicialmente os valores `1, 2, 3, 4, 5`, então os 4 menores elementos são `1, 2, 3, 4`, já os 4 maiores são `2, 3, 4, 5`, logo suas respectivas somas serão `10` e `14`.

Para analisar mais exemplos veja a *Tabela 6.1*.

Tabela 6.1

Entrada	Saída
---------	-------

1, 2, 3, 4, 5	10, 14
0, 2, 6, 3, 4	9, 15
10, 23, 61, 37, 41	9, 15

QUESTÃO 7 - Detecte o elemento de maior frequência de um array (Fonte: Migratory Birds - Hackerrank).

Por exemplo, no caso do array 1, 4, 4, 4, 5, 3 os elementos 1, 3, 4, 5 existem na frequência de 1, 1, 3, 1, logo percebe-se que o elemento 4 possui a maior frequência.

Para analisar mais exemplos veja a *Tabela 7.1*.

Tabela 7.1

Entrada	Saída
1, 4, 4, 4, 5, 3	4
1, 1, 1, 4, 5, 3	1
1, 2, 3, 4, 5, 4, 3, 2, 1, 3, 4	3

QUESTÃO 8 - Crie um algoritmo que receba a string `s` e tenta repeti-la várias vezes até ficar com um tamanho de `n`, e seu retorno deve trazer a quantidade de vezes que aparece a letra `a` (Fonte: Repeated String - Hackerrank).

Por exemplo, se a string `aba` for repetida 4 vezes ela ficará `abaabaabaaba`, mas para ficar com um tamanho de 10 será necessário remover algumas

letras no final, gerando a string `abaabaabaa`. Ao contabilizar a ocorrência da letra `a` o resultado será 7.

Para analisar mais exemplos veja a *Tabela 8.1*.

Tabela 8.1

Entrada	Saída
<code>a, 10</code>	<code>10</code>
<code>b, 10</code>	<code>0</code>
<code>aba, 10</code>	<code>7</code>
<code>a, 10000000000000</code>	<code>10000000000000</code>

QUESTÃO 9 - Retornar a sequência linear de `a` até `b`, e em seguida, a sequência será espelhada (Fonte: Mirror Sequence - URI Online Judge).

Por exemplo, a sequência de `1` até `5` é `12345`, e seu reflexo é `54321`, juntando os dois segmento obtem-se o valor `1234554321`.

Para analisar mais exemplos veja a *Tabela 9.1*.

Tabela 9.1

Entrada	Saída
<code>1, 5</code>	<code>1234554321</code>
<code>10, 13</code>	<code>1011121331211101</code>
<code>98, 101</code>	<code>98991001011010019989</code>

QUESTÃO 10 - Retonar a soma dos valores `a` e `b` sem o dígito `0` caso exista (Fonte: Zero means Zero - URI Online Judge).

Por exemplo, a soma de 7 mais 8 é igual a 15, logo o retorno será 15 pois não existe o dígito `0`. Já a soma de 15 mais 5 é igual a 20, logo o retorno será 2 pois será removido o dígito `0`.

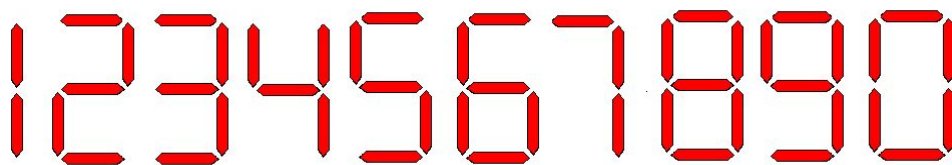
Para analisar mais exemplos veja a *Tabela 10.1*.

Tabela 10.1

Entrada	Saída
7, 8	15
15, 5	2
99, 6	15

QUESTÃO 11 - Receber um número e retorno o total de LEDs necessário para exibi-lo. Lembrando que conforme a *Figura 11.1* exhibe, cada número exige uma quantidade de LEDs específica (Fonte: Led - URI Online Judge).

Figura 11.1 - Representação dos números em LED



(Fonte: Led - URI Online Judge)

Então, seguindo o raciocínio, o dígito `1` precisa de 2 LEDs, o `2` de 5 e assim sucessivamente. Já o número `115380` é formado pelos dígitos `{1, 1, 5,`

3, 8, 0} e exige respectivamente a seguinte quantia de LEDs {2, 2, 5, 5, 7, 6}, que resulta em um total de 27 LEDs.

Para analisar mais exemplos veja a *Tabela 11.1*.

Tabela 11.1

Entrada	Saída
1	2
2	5
23456	25
115380	27
2819311	29