

<p>1. O programa deve ser modularizado e a saída formatada com mensagens explicativas. Usar Try para validar os dados. Ler uma quantidade indeterminada de nomes que será associado a uma data gerada aleatoriamente do tipo dd/mm/aaaa. Os nomes devem ser validados para caracteres alfabéticos ou espaço em branco e as datas devem ser compatíveis com dias e meses do ano. Anos entre 2000 e 2018. Classificar e exibir em ordem ascendente por nome. Exibir o nome e a data Não pode usar sort().</p>
<p>2. O programa deve ser modularizado e a saída formatada com mensagens explicativas. Usar Try para validar os dados. Ler uma quantidade indeterminada de nomes que será associado a uma data gerada aleatoriamente do tipo dd/mm/aaaa. Os nomes devem ser validados para caracteres alfabéticos ou espaço em branco e as datas devem ser compatíveis com dias e meses do ano. Anos entre 2000 e 2018. Classificar e exibir em ordem ascendente por data. Exibir o nome e a data Não pode usar sort(). Pode usar replace()</p>
<p>3. O programa deve ser modularizado e a saída formatada com mensagens explicativas. Usar Try para validar os dados. Ler uma quantidade indeterminada de nomes que será associado a um numero de matricula de 6 algarismos gerados aleatoriamente. O número de matricula é único e deve ser validado quanto a repetição. Os nomes devem ser validados para caracteres alfabéticos ou espaço em branco . Usando pesquisa binária verificar se uma determinada matricula (apenas a matricula é lida) se encontra na lista. A resposta deverá exibir nome e matricula e a mensagem “foi encontrado” ou “não foi encontrado” Pode usar sort() e replace()</p>
<p>4. O programa deve ser modularizado e a saída mostrada com mensagens explicativas Try para validar os dados Gerar valores aleatórios e reais compreendidos entre 1 e 99 para uma matriz de mxn elementos. Os valores de m e n são lidos. Determinar o vetor soma da 1ª. Linha com a 2ª. Linha. Exibir a matriz lida sob a forma de tabela e o vetor soma. Os valores devem ser exibidos com 2 casas decimais.</p>
<p>5. O programa deve ser modularizado e a saída formatada com mensagens explicativas. Usar Try para validar os dados. Ler uma quantidade indeterminada de nomes que será associado a um numero de matricula de 6 algarismos gerados aleatoriamente. O número de matricula é único e deve ser validado quanto a repetição. Os nomes devem ser validados para caracteres alfabéticos ou espaço em branco . Usando pesquisa binária verificar se um determinado nome (apenas o nome procurado é lido) se encontra na lista lida. A resposta deverá exibir nome e matricula e a mensagem “foi encontrado” ou “não foi encontrado” Pode usar sort() e replace()</p>

6. O programa deve ser modularizado e a saída formatada com mensagens explicativas.

Usar Try para validar os dados.

Ler uma quantidade indeterminada de matriculas cada qual será associada a um nome de 5 a 10 letras gerados aleatoriamente. O número de matricula é único e deve ser validado quanto a repetição.

Sabe-se que os caracteres ASCII que definem letras minúsculas são entre 97 (letra a) e 122 (letra z)

Usando pesquisa binária verificar se uma determinada matricula (apenas a matricula procurada é lida) se encontra na lista lida.

A resposta deverá exibir nome e matricula e a mensagem “foi encontrado” ou “não foi encontrado”

Pode usar sort() e replace()

7. O programa deve ser modularizado e a saída formatada com mensagens explicativas.

Gerar aleatoriamente ‘n’ nomes com duas palavras de 5 a 10 letras cada. Os nomes podem ser criados usando-se apenas letras minúsculas.

Sabe-se que os caracteres ASCII que definem letras minúsculas são entre 97 (letra a) e 122 (letra z) e espaço em branco é o código 32.

Usando short bubble classificar os nomes em ordem alfabética.

Exibir os nomes gerados originalmente e depois ordenados.

8. O programa deve ser modularizado e a saída formatada com mensagens explicativas.

Gerar aleatoriamente ‘n’ nomes com duas palavras de 5 a 10 letras cada. Os nomes devem ter a primeira letra maiúsculas, as demais devem ser apenas letras minúsculas.

Sabe-se que os caracteres ASCII que definem letras minúsculas são entre 97 (letra a) e 122 (letra z) e espaço em branco é o código 32.

Usando bubble sort classificar os nomes em ordem alfabética inversa (de z para a).

Exibir os nomes gerados originalmente e depois ordenados.

9. O programa deve ser modularizado e a saída formatada com mensagens explicativas.

Gerar números aleatórios entre 1 e 99 para dois vetores A e B de ‘m’ elementos cada. Construir uma matriz C de duas dimensões e ‘m’ elementos com a seguinte lei de formação: a primeira linha de C são os elementos de A em ordem ascendente. A segunda linha de C são os elementos de B em ordem descendente.

Exibir A,B e C com legenda/ mensagens explicativas.

O valor de ‘m’ é lido

10. O programa deve ser modularizado e a saída formatada com mensagens explicativas

Elabore um programa para gerar dados inteiros e aleatórios entre 1 e 8 para um vetor A de uma dimensão com ‘m’ elementos inteiros. Construir uma matriz B de duas dimensões com 3 colunas, onde a primeira coluna da matriz B é formada pelos elementos da matriz A somados com mais 5, a segunda coluna é formada pelo valor do calculo do fatorial de cada elemento correspondente da matriz A e a terceira coluna deverá ser formada pelos quadrados dos elementos correspondentes da matriz A. Exibir o vetor A ,bem como a matriz B..