

Dredd - Juiz Online

Principal

Perfil

Minhas Provas

Sair

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

REO 4 - ABB, AVL e Árvore Rubro-Negra

Prova Aberta Até: 30/08/2022 23:59:59

Número Máximo de Tentativas: 3

Atenuação da Nota por Tentativa: 0%

Instruções para a prova: A lista é individual. Você pode trocar ideias com colegas, desde que não copie o código.

Questão 1: Contagem dos nós folha de uma Árvore Binária de Busca (ABB)

A partir do [código fornecido](#), implemente os métodos que calculam a quantidade de nós folhas em uma árvore binária. Não é necessário manter o projeto do programa exemplo. Métodos auxiliares podem ser criados. Métodos da contagem de folhas que não forem usados devem ser apagados.

A quantidade de folhas numa árvore vazia é zero.

Entradas:

1. Número de valores a serem lidos.
2. Valores (inteiros) a inserir na árvore.

Saídas:

- Quantidade de nós folha.

Exemplo de Entrada:

8
30 15 50 41 10 60 32 45

Exemplo de Saída:

4

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 2: Encontre o sucessor em uma Árvore Binária de Busca (ABB)

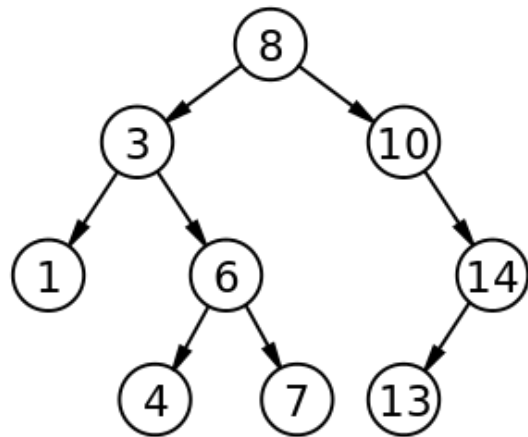
Minutos Restantes:
?

Usuário:
Lucas Gomes Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Implemente um método para encontrar o sucessor de uma chave numa árvore binária. Suponha que a chave pertence à árvore. É importante que haja um mínimo de eficiência: buscar a chave (já pronto) e depois caminhar até o sucessor, sem processar outros nós.



Use [esta implementação](#) que já tem pronto método para escrever a árvore, além de uma interface que ativa os métodos implementados. A interface não deve ser alterada.

Antes de implementar, planeje. Note que um nó que não tem filho à direita tem um sucessor que não está numa subárvore. Na árvore da figura, por exemplo, o sucessor de 7 está na raiz.

Entradas:

O programa lê comandos e os executa. Os sucessores são encontrados chamando repetidamente o método que encontra um sucessor. Basta usar a interface providenciada. Os comandos são:

- f - para finalizar a execução do programa
- i - para inserir uma chave, seguido da chave (inteiro)
- e - para escrever o conteúdo da árvore, nível a nível para facilitar o entendimento da estrutura
- s - para escrever todos os sucessores de uma chave, seguido da chave (inteiro)

Saídas:

Cada comando produz uma saída específica. Apenas mantenha as saídas já implementadas.

Exemplo de entrada e saída juntos:

```

i 300
i 20
i 430
i 12
i 360
i 451
s 20
300 360 430 451
f
  
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo escolhido

Questão 3: Árvore binária (ABB) - Árvore de Palavras

Considere um arquivo texto preenchido de nome entrada.txt. Esse arquivo deve ser o parâmetro de entrada para o seu sistema que deve:

- armazenar cada palavra do texto em um nó de uma árvore binária de busca
- armazenar, para cada palavra, a posição em que ela aparece no texto
- permitir a busca por uma determinada palavra. Caso a palavra esteja presente na árvore, o programa deverá imprimir as posições em que se encontra. Caso a palavra não seja encontrada, o programa deve imprimir -1

Exemplo:

E agora, Jose?

A festa acabou,

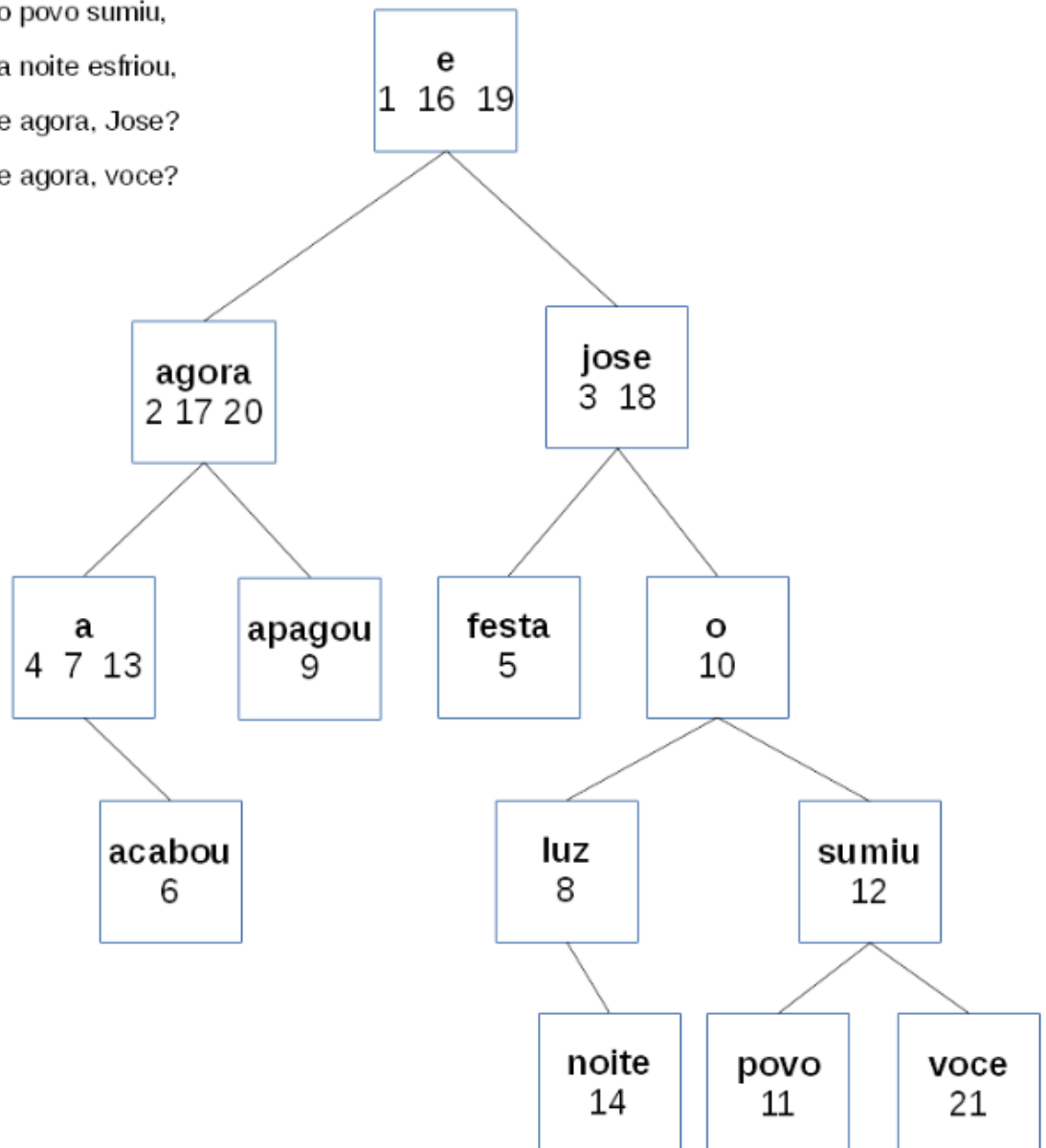
a luz apagou,

o povo sumiu,

a noite esfriou,

e agora, Jose?

e agora, voce?



Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Neste exemplo a palavra **e** aparece nas posições 1, 16 e 19, ou seja, ela aparece mais de uma vez no texto, mas deve ser representada por apenas um nó na árvore. As posições das repetições devem ser armazenadas no mesmo nó.

Observação: Considere que o texto presente no arquivo já passou por um tratamento, assim já foram removidos acentos e pontuação além de todas as letras serem minúsculas.

Entradas:

1. Arquivo de entrada
2. Palavra a ser buscada

Saídas:

1. Posições da palavra

Exemplo de Entrada:

No arquivo:

e agora jose
a festa acabou
a luz apagou
o povo sumiu
a noite esfriou
e agora jose
e agora voce

Na entrada padrão:

agora

Exemplo de Saída:

2 17 20

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo escolhido

Questão 4: Árvore binária (ABB) - Contagem de nós

Faça um programa que calcula a quantidade de nós em uma árvore binária. Seu programa também deve exibir a diferença de nós da sub-árvore esquerda com a sub-árvore direita da raiz (esquerda menos direita). A leitura dos valores deve ser feita até que o valor -1 seja lido (ele representará que a entrada acabou).

A quantidade de nós numa árvore vazia é zero.

Não é permitido usar atributos que indiquem quantidade de nós na árvore e nem no nó.

Entradas:

1. Valores (inteiros) a inserir na árvore.

Saídas:

1. Quantidade de nós da árvore.
2. Diferença de nós da sub-árvore esquerda com a sub-árvore direita da raiz.

Exemplo de Entrada:

45 30 10 69 5 187 100 50 25 2 -1

Exemplo de Saída:

10 1

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Peso: 1

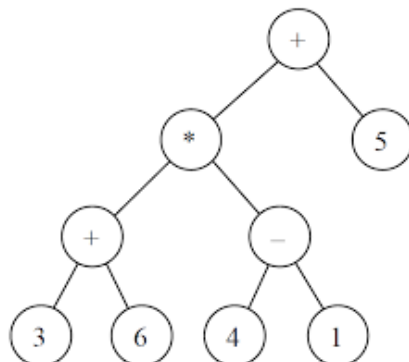
Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo escolhido

Questão 5: Árvores Binárias - Resolução de Expressões em Notação Prefixa

Em várias áreas da computação, árvores são utilizadas para resolver vários problemas e aplicadas de forma eficiente para estruturação de "registros" seguindo critérios específicos. Entre as várias aplicações de árvores binárias, encontram-se as árvores de expressões. Em uma árvore de expressões, expressões lógicas ou aritméticas podem ser representadas sem ambiguidade e sem a necessidade de uso de parênteses. Assim, a imagem apresentada ilustra, sem ambiguidade e sem parênteses, a expressão $((3+6)*(4-1))+5$. Computacionalmente, isso é importante, principalmente no desenvolvimento de compiladores e interpretadores e está relacionado ao uso de notações prefixas ou posfixas de expressões.



Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Em uma representação prefixa, também chamada de notação polonesa, as expressões são representadas com operadores vindo antes dos operandos. Por exemplo, a expressão **(a+b)** seria representada como **+ a b**. O termo notação polonesa deve-se à sua criação pelo estudioso de lógica polonês Jan Łukasiewicz. Perceba que a notação polonesa corresponde justamente ao percorrimto prefixo da árvore de expressões. Perceba também, que na árvore de expressões, os operadores são nós intermediários e os valores ou variáveis numéricas são nós folhas.

Seu objetivo é escrever um programa que aceite uma expressão aritmética escrita em notação prefixa e construir a árvore de expressão associada, efetuando o cálculo da expressão usando o percorrimto adequado à tarefa.

Dica: use `stoi()` para converter strings para valores inteiros.

Entradas:

- Expressão aritmética em notação polonesa, composta pelos caracteres: `+`, `-`, `*`, `/`, `^`, em que `^` indica potenciação e os outros operadores indicam as mesmas operações presentes em C++. Considere que a expressão informada sempre será válida.

Saídas:

- Resultado da operação após manipulação da árvore. Em casos de operações inválidas, como divisão por zero, deverá ser escrito **invalido** na saída padrão.

Exemplo de Entrada:

`+ * + 3 6 - 4 1 5`

Exemplo de Saída:

`32`

Exemplo de Entrada:

`- * 2 / 8 4 3`

Exemplo de Saída:

`1`

Exemplo de Entrada:

`+ ^ 3 2 / * 2 2 - 4 3`

Exemplo de Saída:

`13`

Exemplo de Entrada:

`+ ^ 3 2 / * 2 2 - 4 4`

Exemplo de Saída:

`invalido`

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?

Q2: ?

Q3: ?

Q4: ?

Q5: ?

Q6: ?

Q7: ?

Q8: ?

Q9: ?

Q10: ?

Q11: ?

Q12: ?

Q13: ?

Q14: ?

Q15: ?

Q16: ?

Q17: ?

Total: 0

Questão 6: Árvore Binária com Contadores

Implemente a árvore binária com contadores, de forma que ela permita elementos repetidos, mas que as repetições sejam armazenadas no próprio nó. Ou seja, cada nó conterá, além do valor, um contador para informar quantos elementos daquele valor estão armazenados. Ao inserir um elemento na árvore, caso já exista um nó com esse valor, então o contador é incrementado. O processo de remoção, por sua vez, irá decrementar o contador, só removendo o nó quando o valor for totalmente excluído da árvore. Você deverá implementar os métodos de inserção, remoção e de percorrimto. A função de percorrimto deverá imprimir, logo após o valor, a quantidade de elementos armazenados naquele nó. Por exemplo, a representação 1(4)/0 indica que o valor 1 possui 4 elementos armazenados e encontra-se no nível 0 da árvore.

Entradas:

1. 10 valores a serem inseridos na árvore
2. 5 valores a serem removidas da árvore
3. 10 valores a serem inseridos na árvore
4. 5 valores a serem removidas da árvore

Saídas:

1. Percorrimento da árvore após 10 inserções
2. Percorrimento da árvore após 5 remoções
3. Percorrimento da árvore após 10 inserções
4. Percorrimento da árvore após 5 remoções

Exemplo de Entrada:

```
1 1 1 1 2 2 2 3 3 3
1 1 2 2 2
4 4 5 5 5 2 2 6 6 6
5 5 6 1 1
```

Exemplo de Saída:

```
1(4)/0 2(3)/1 3(3)/2
1(2)/0 3(3)/1
1(2)/0 2(2)/2 3(3)/1 4(2)/2 5(3)/3 6(3)/4
2(2)/1 3(3)/0 4(2)/1 5(1)/2 6(2)/3
```

Exemplo de Entrada:

```
1 1 1 1 2 2 2 3 3 3
1 1 2 2 3
1 1 1 1 1 2 2 2 2 2
3 3 1 1 1
```

Exemplo de Saída:

```
1(4)/0 2(3)/1 3(3)/2
1(2)/0 2(1)/1 3(2)/2
1(7)/0 2(6)/1 3(2)/2
1(4)/0 2(6)/1
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

 Nenhum arquivo escolhido

Questão 7: Implementação de uma Árvore Binária de Busca (ABB) sem nó

Existem muitas formas de implementar uma estrutura de dados qualquer. A linguagem de programação usada é um dos fatores que mais influência as características de implementação. As linguagens que escondem ponteiros e usam *coleta de lixo* para gerenciar o uso da memória favorecem a implementação de árvores sem a classe auxiliar "nó".

Linguagens sem *coleta de lixo* não favorecem esse tipo de implementação, mas isso não quer dizer que não podemos fazer uma.

Implemente uma ABB sem usar uma classe auxiliar. Para agilizar seu desenvolvimento, use [código fornecido](#) que já tem atributos e métodos projetados, além de um programa com interface para inserir, buscar, remover e escrever a árvore. Os métodos auxiliares no código são sugestão de implementação. O programa lê comandos numéricos e seus argumentos para as várias operações possíveis.

Entradas:

Cada comando é um número inteiro identificando o comando seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- O número 0 para encerrar a execução do programa.
- O número 1 para inserir chave (número inteiro) e valor (número inteiro) na árvore
- O número 2 para remover dado da árvore, seguido da chave (número inteiro) que deve ser removida.
- O número 3 para buscar na árvore, seguido da chave consultada. Este comando produz uma saída que é o valor associado.
- O número 4 para escrever todas chaves e seus respectivos valores num formato de texto com parênteses.
- O número 5 para escrever os nós da árvore nível a nível.

Saídas:

Cada comando tem sua saída específica. Veja os exemplos abaixo.

Exemplo de entrada e saída juntos:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
1 5 50
1 3 30
1 2 20
4
(5/50 (3/30 (2/20 () ()) ()) ())
1 6 60
5
[5/50]
[3/30][6/60]
[2/20][][][]
[][]
2 5
5
[6/60]
[3/30][]
[2/20][]
[][]
2 4
Impossível remover. A chave não existe.
2 3
4
(6/60 (2/20 () ()) ())
2 2
2 6
5
[]
0
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 8: Árvore ABB - Nó sem pai

Em algumas implementações de árvores binárias de busca, os nós têm encadeamento duplo (para baixo e para cima) enquanto que em outras eles tem encadeamento simples (só para baixo). A primeira estratégia simplifica a implementação de alguns métodos enquanto a segunda economiza memória.

Complete o **código fornecido** para produzir uma implementação **com encadeamento simples**. A implementação deve possuir as operações de inserção e busca. Se quiser, implemente uma operação de escrever a árvore para facilitar a depuração do programa. Ela não será usada na avaliação do programa.

A função principal foi feita supondo o uso de manipulação de exceções. Você pode alterar essa estratégia, desde que a operação de escrita da mensagem de erro continue na função principal e continue sendo realizada na saída padrão.

Entradas:

1. i - para **inserir** uma informação na árvore. A letra i deve ser seguida da chave (número inteiro) e valor (palavra).

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

2. b - para **buscar** uma informação na árvore. A letra b deve ser seguida da chave a ser buscada.
3. e - para escrever o conteúdo da árvore, em ordem.
4. f - para finalizar a execução do programa.

Saídas:

A operação de inserção não produz saída. A operação de busca escreve o valor associado à chave. Caso a informação buscada não esteja na árvore, a função principal escreve "INEXISTENTE" no lugar do valor. A operação de escrever não tem validade na avaliação.

Exemplo de Entrada:

```
i 4 quatro
i 2 dois
i 8 oito
i 6 seis
b 2
b 6
b 7
f
```

Exemplo de Saída:

```
dois
seis
INEXISTENTE
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo escolhido

Questão 9: Estrutura de dados AVL recursiva - ONG de Adoção de Animais

Uma ONG para adoção de animais resolveu catalogar os animais sob seu cuidado. Para isso, cada animal será registrado com seu nome, espécie e raça. Utilizando o [código fornecido](#), implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **nome do animal** do tipo *string*, a **espécie** (**gato, cachorro, etc.**) do animal do tipo *string*, **raça** (**pequês, vira-lata, etc.**) do animal do tipo *string*.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , nome do animal [string], espécie [string], raça [string]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** de quantidade de animais de uma dada espécie e raça: deve ler duas strings (espécie e raça)
5. e - para **imprimir** árvore
6. f - finalizar

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([chave],[nome],[espécie],[raça])", em que [chave], [nome], [espécie] e [raça] devem ser substituídos pela respectiva chave, nome do animal, sua espécie e raça
4. l - levantamento por localização: Imprime: "Levantamento de animais: [quantidade]", em que [quantidade] é número de animais de uma dada espécie e raça.
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 5 doguo cao viralata
i 10 mieco gato persa
i 15 tanin gato persa
i 3 xicao cao viralata
i 2 pepe cao pequines
i 1 popo cao pequines
i 200 drum cao fila
i 201 bitao cao viralata
i 399 bigorna cao pequines
i 12 baran gato viralata
i 31 xaxau gato persa
i 32 xupita cao pequines
e
l cao pequines
l cao russo
l gato persa
f
```

Exemplo de Saída:

```
(10,mieco)
├─e-(3,xicao)
│   └─e-(2,pepe)
│       └─e-(1,popo)
├─d-(5,doguo)
├─d-(200,drum)
├─e-(15,tanin)
│   └─e-(12,baran)
│       └─d-(31,xaxau)
│           └─d-(32,xupita)
├─d-(201,bitao)
│   └─d-(399,bigorna)
Levantamento de animais: 4
Levantamento de animais: 0
Levantamento de animais: 3
```

Exemplo de Entrada 2:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
i 5 doguo cao viralata
i 10 mieco gato persa
i 15 tanin gato persa
i 3 xicao cao viralata
i 2 pepe cao pequines
i 1 popo cao pequines
i 200 drum cao fila
i 201 bitao cao viralata
i 399 bigorna cao pequines
i 12 baran gato viralata
i 31 xaxau gato persa
i 32 xupita cao pequines
e
r 1
r 2
r 10
i 1 popo cao fila
i 2 pepe cao russo
i 10 mieco gato siames
e
l cao pequines
l cao russo
l gato persa
f
```

Exemplo de Saída 2:

```
(10,mieco)
  e-(3,xicao)
    e-(2,pepe)
      e-(1,popo)
    d-(5,doguo)
  d-(200,drum)
    e-(15,tanin)
      e-(12,baran)
        d-(31,xaxau)
          d-(32,xupita)
      d-(201,bitao)
        d-(399,bigorna)
(15,tanin)
  e-(5,doguo)
    e-(2,pepe)
      e-(1,popo)
        d-(3,xicao)
    d-(12,baran)
      e-(10,mieco)
  d-(200,drum)
    e-(31,xaxau)
      d-(32,xupita)
    d-(201,bitao)
      d-(399,bigorna)
Levantamento de animais: 2
Levantamento de animais: 1
Levantamento de animais: 2
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?

Q2: ?

Q3: ?

Q4: ?

Q5: ?

Q6: ?

Q7: ?

Q8: ?

Q9: ?

Q10: ?

Q11: ?

Q12: ?

Q13: ?

Q14: ?

Q15: ?

Q16: ?

Q17: ?

Total: 0

Questão 10: Estrutura de dados AVL recursiva - Encontro de Motociclistas

A Associação Duas Rodas está organizando um encontro de motociclistas. Os responsáveis pelo encontro decidiram fazer uma série de levantamentos de dados para, a partir disso, organizar um encontro memorável. Assim eles, irão cadastrar todas as motocicletas que participarão do evento, armazenando para isso o modelo da motocicleta, sua marca e o ano de fabricação. Utilizando o **código fornecido** implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor). Além das funções tradicionais, eles irão também implementar um método para levantamento da quantidade de motocicletas de uma dada marca que tenham sido fabricadas após um determinado ano.

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **modelo da motocicleta** do tipo *string*, a **marca da motocicleta** do tipo *string* e o **ano de fabricação** do tipo *inteiro*. Não existem duas motocicletas com mesma chave no sistema.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , modelo da motocicleta [string], marca da motocicleta [string] e o ano de fabricação[inteiro]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** da quantidade de motocicletas de uma mesma marca fabricadas a partir de um dado ano
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([modelo],[marca],[ano])", em que [modelo], [marca] e [ano] devem ser substituídos pelo respectivo modelo, marca e ano de fabricação da motocicleta
4. l - levantamento por ano: Imprime: "Levantamento da [marca]: [quantidade]", em que [marca] é a marca de motocicleta informada para busca e [quantidade] é o número de motocicletas fabricadas a partir do ano (também informado para o levantamento)
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
i 80 PBR Arruela 1998
i 3 YBR IrNaMarra 1980
i 5 TXT Ronda 1990
i 9 ZTX Rarledavisson 1993
i 11 SZK Suzuki 1998
i 160 PBZ Arruela 2002
i 15 BZB Bizon 2001
e
l IrNaMarra 1970
i 8 YBZ IrNaMarra 1990
i 22 YBW IrNaMarra 1995
l IrNaMarra 1990
l IrNaMarra 1970
e
f
```

Exemplo de Saída:

```
(11,SZK)
├─e-(5,TXT)
│   └─e-(3,YBR)
│       └─d-(9,ZTX)
└─d-(80,PBR)
    └─e-(15,BZB)
        └─d-(160,PBZ)
Levantamento da marca IrNaMarra: 1
Levantamento da marca IrNaMarra: 2
Levantamento da marca IrNaMarra: 3
(11,SZK)
├─e-(5,TXT)
│   └─e-(3,YBR)
│       └─d-(9,ZTX)
│           └─e-(8,YBZ)
└─d-(80,PBR)
    └─e-(15,BZB)
        └─d-(22,YBW)
            └─d-(160,PBZ)
```

Exemplo de Entrada:

```
i 80 PBR Arruela 1998
i 3 YBR IrNaMarra 1980
i 5 TXT Ronda 1990
i 9 ZTX Rarledavisson 1993
i 11 SZK Suzuki 1998
i 160 PBZ Arruela 2002
i 15 BZB Bizon 2001
i 8 YBZ IrNaMarra 1990
i 22 YBW IrNaMarra 1995
e
r 3
r 80
r 11
e
l IrNaMarra 1990
l IrNaMarra 1970
f
```

Exemplo de Saída:

```
(11,SZK)
├─e-(5,TXT)
└─e-(3,YBR)
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```

      Ld-(9,ZTX)
      |
      Le-(8,YBZ)
    Ld-(80,PBR)
      |
      Le-(15,BZB)
      |
      Ld-(22,YBW)
      |
      Ld-(160,PBZ)
(15,BZB)
  |
  Le-(8,YBZ)
  |
  Le-(5,TXT)
  |
  Ld-(9,ZTX)
  |
  Ld-(22,YBW)
  |
  Ld-(160,PBZ)
Levantamento da marca IrNaMarra: 2
Levantamento da marca IrNaMarra: 2

```

Exemplo de Entrada:

```

i 80 PBR Arruela 1998
i 3 YBR IrNaMarra 1980
i 5 TXT Ronda 1990
i 9 ZTX Rarledavisson 1993
i 11 SZK Suzuki 1998
i 160 PBZ Arruela 2002
i 15 BZB Bizon 2001
i 8 YBZ IrNaMarra 1990
i 22 YBW IrNaMarra 1995
e
r 3
r 80
r 11
b 11
i 11 SZZ Suzuki 1998
i 80 PBRt Arruela 1998
i 3 YBRX IrNaMarra 1980
e
b 11
f

```

Exemplo de Saída:

```

(11,SZK)
  |
  Le-(5,TXT)
  |
  Le-(3,YBR)
  |
  Ld-(9,ZTX)
  |
  Le-(8,YBZ)
  |
  Ld-(80,PBR)
  |
  Le-(15,BZB)
  |
  Ld-(22,YBW)
  |
  Ld-(160,PBZ)
Erro na busca: elemento não encontrado!
(15,BZB)
  |
  Le-(8,YBZ)
  |
  Le-(5,TXT)
  |
  Le-(3,YBRX)
  |
  Ld-(9,ZTX)
  |
  Ld-(11,SZZ)
  |
  Ld-(80,PBRt)
  |
  Le-(22,YBW)
  |
  Ld-(160,PBZ)
Elemento buscado: (11,SZZ,Suzuki,1998)

```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo Nenhum arquivo escolhido

Enviar Resposta

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Questão 11: Estrutura de dados AVL recursiva - Floricultura

Uma floricultura resolveu catalogar o seu estoque de sementes. Para isso, cada pacote de semente será registrado com nome do cultivar, marca da semente, ano de fabricação e a quantidade de itens disponíveis para venda. Utilizando o **código fornecido**, implemente um **árvore AVL** com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinatura das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **nome do cultivar** do tipo *string*, a **marca da semente** do tipo *string*, **ano de fabricação** do tipo *inteiro* e a **quantidade de itens** disponíveis do tipo *inteiro*.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , nome do cultivar [string], marca [string], ano de fabricação [inteiro] e quantidade disponível [inteiro]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** de quantidade de pacotes de sementes de um cultivar de uma dada marca: deve ler duas strings (cultivar e marca)
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([chave],[cultivar],[marca],[ano],[quantidade])", em que [chave], [nome], [local], [ano] e [quantidade] devem ser substituídos pela respectiva chave, nome do cultivar, marca da semente, ano de fabricação e quantidade disponível
4. l - levantamento por localização: Imprime: "Levantamento do cultivar [cultivar]: [quantidade]", em que [cultivar] é o cultivar pesquisado e [quantidade] é a quantidade de pacotes de sementes daquele cultivar com uma dada marca

5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 4 tomate astrix 2008 5
i 5 tomate panun 2009 6
i 8 rosa astrix 2010 3
i 10 rosa panu 2013 2
i 101 cravo astrix 2018 3
i 201 tomate astrix 2010 5
i 301 tomate astrix 2015 10
i 15 manjericao tafum 2018 3
i 18 manjericao tafum 2020 5
e
l manjericao tafum
l tomate astrix
f
```

Exemplo de Saída:

```
(10,rosa)
├──e-(5,tomate)
│   ├──e-(4,tomate)
│   └──d-(8,rosa)
└──d-(201,tomate)
    ├──e-(18,manjericao)
    │   └──e-(15,manjericao)
    │       └──d-(101,cravo)
    └──d-(301,tomate)
```

Levantamento do cultivar manjericao: 8
Levantamento do cultivar tomate: 20

Exemplo de Entrada 2:

```
i 4 tomate astrix 2008 5
i 5 tomate panun 2009 6
i 8 rosa astrix 2010 3
i 10 rosa panu 2013 2
i 101 cravo astrix 2018 3
i 201 tomate astrix 2010 5
i 301 tomate astrix 2015 10
i 15 manjericao tafum 2018 3
i 18 manjericao tafum 2020 5
e
b 201
b 142
l rosa panun
r 8
r 10
r 301
i 10 rosa panun 2013 2
i 303 rosa panun 2014 3
e
l rosa panun
f
```

Exemplo de Saída 2:

```
(10,rosa)
|e-(5,tomate)
|   |e-(4,tomate)
```

Minutos Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?

Q2: ?

Q3: ?

Q4: ?

Q5: ?

Q6: ?

Q7: ?

Q8: ?

Q9: ?

Q10: 'Q11: 'Q12: 'Q13: 'Q14: 'Q15: 'Q16: 'Q17: 'Q18: 'Q19: 'Q20: 'Q21: 'Q22: 'Q23: 'Q24: 'Q25: 'Q26: 'Q27: 'Q28: 'Q29: 'Q30: 'Q31: 'Q32: 'Q33: 'Q34: 'Q35: 'Q36: 'Q37: 'Q38: 'Q39: 'Q40: 'Q41: 'Q42: 'Q43: 'Q44: 'Q45: 'Q46: 'Q47: 'Q48: 'Q49: 'Q50: 'Q51: 'Q52: 'Q53: 'Q54: 'Q55: 'Q56: 'Q57: 'Q58: 'Q59: 'Q60: 'Q61: 'Q62: 'Q63: 'Q64: 'Q65: 'Q66: 'Q67: 'Q68: 'Q69: 'Q70: 'Q71: 'Q72: 'Q73: 'Q74: 'Q75: 'Q76: 'Q77: 'Q78: 'Q79: 'Q80: 'Q81: 'Q82: 'Q83: 'Q84: 'Q85: 'Q86: 'Q87: 'Q88: 'Q89: 'Q90: 'Q91: 'Q92: 'Q93: 'Q94: 'Q95: 'Q96: 'Q97: 'Q98: 'Q99: 'Q100: 'Q101: 'Q102: 'Q103: 'Q104: 'Q105: 'Q106: 'Q107: 'Q108: 'Q109: 'Q110: 'Q111: 'Q112: 'Q113: 'Q114: 'Q115: 'Q116: 'Q117: 'Q118: 'Q119: 'Q120: 'Q121: 'Q122: 'Q123: 'Q124: 'Q125: 'Q126: 'Q127: 'Q128: 'Q129: 'Q130: 'Q131: 'Q132: 'Q133: 'Q134: 'Q135: 'Q136: 'Q137: 'Q138: 'Q139: 'Q140: 'Q141: 'Q142: 'Q143: 'Q144: 'Q145: 'Q146: 'Q147: 'Q148: 'Q149: 'Q150: 'Q151: 'Q152: 'Q153: 'Q154: 'Q155: 'Q156: 'Q157: 'Q158: 'Q159: 'Q160: 'Q161: 'Q162: 'Q163: 'Q164: 'Q165: 'Q166: 'Q167: 'Q168: 'Q169: 'Q170: 'Q171: 'Q172: 'Q173: 'Q174: 'Q175: 'Q176: 'Q177: 'Q178: 'Q179: 'Q180: 'Q181: 'Q182: 'Q183: 'Q184: 'Q185: 'Q186: 'Q187: 'Q188: 'Q189: 'Q190: 'Q191: 'Q192: 'Q193: 'Q194: 'Q195: 'Q196: 'Q197: 'Q198: 'Q199: 'Q200: 'Q201: 'Q202: 'Q203: 'Q204: 'Q205: 'Q206: 'Q207: 'Q208: 'Q209: 'Q210: 'Q211: 'Q212: 'Q213: 'Q214: 'Q215: 'Q216: 'Q217: 'Q218: 'Q219: 'Q220: 'Q221: 'Q222: 'Q223: 'Q224: 'Q225: 'Q226: 'Q227: 'Q228: 'Q229: 'Q230: 'Q231: 'Q232: 'Q233: 'Q234: 'Q235: 'Q236: 'Q237: 'Q238: 'Q239: 'Q240: 'Q241: 'Q242: 'Q243: 'Q244: 'Q245: 'Q246: 'Q247: 'Q248: 'Q249: 'Q250: 'Q251: 'Q252: 'Q253: 'Q254: 'Q255: 'Q256: 'Q257: 'Q258: 'Q259: 'Q260: 'Q261: 'Q262: 'Q263: 'Q264: 'Q265: 'Q266: 'Q267: 'Q268: 'Q269: 'Q270: 'Q271: 'Q272: 'Q273: 'Q274: 'Q275: 'Q276: 'Q277: 'Q278: 'Q279: 'Q280: 'Q281: 'Q282: 'Q283: 'Q284: 'Q285: 'Q286: 'Q287: 'Q288: 'Q289: 'Q290: 'Q291: 'Q292: 'Q293: 'Q294: 'Q295: 'Q296: 'Q297: 'Q298: 'Q299: 'Q300: 'Q301: 'Q302: 'Q303: 'Q304: 'Q305: 'Q306: 'Q307: 'Q308: 'Q309: 'Q310: 'Q311: 'Q312: 'Q313: 'Q314: 'Q315: 'Q316: 'Q317: 'Q318: 'Q319: 'Q320: 'Q321: 'Q322: 'Q323: 'Q324: 'Q325: 'Q326: 'Q327: 'Q328: 'Q329: 'Q330: 'Q331: 'Q332: 'Q333: 'Q334: 'Q335: 'Q336: 'Q337: 'Q338: 'Q339: 'Q340: 'Q341: 'Q342: 'Q343: 'Q344: 'Q345: 'Q346: 'Q347: 'Q348: 'Q349: 'Q350: 'Q351: 'Q352: 'Q353: 'Q354: 'Q355: 'Q356: 'Q357: 'Q358: 'Q359: 'Q360: 'Q361: 'Q362: 'Q363: 'Q364: 'Q365: 'Q366: 'Q367: 'Q368: 'Q369: 'Q370: 'Q371: 'Q372: 'Q373: 'Q374: 'Q375: 'Q376: 'Q377: 'Q378: 'Q379: 'Q380: 'Q381: 'Q382: 'Q383: 'Q384: 'Q385: 'Q386: 'Q387: 'Q388: 'Q389: 'Q390: 'Q391: 'Q392: 'Q393: 'Q394: 'Q395: 'Q396: 'Q397: 'Q398: 'Q399: 'Q400: 'Q401: 'Q402: 'Q403: 'Q404: 'Q405: 'Q406: 'Q407: 'Q408: 'Q409: 'Q410: 'Q411: 'Q412: 'Q413: 'Q414: 'Q415: 'Q416: 'Q417: 'Q418: 'Q419: 'Q420: 'Q421: 'Q422: 'Q423: 'Q424: 'Q425: 'Q426: 'Q427: 'Q428: 'Q429: 'Q430: 'Q431: 'Q432: 'Q433: 'Q434: 'Q435: 'Q436: 'Q437: 'Q438: 'Q439: 'Q440: 'Q441: 'Q442: 'Q443: 'Q444: 'Q445: 'Q446: 'Q447: 'Q448: 'Q449: 'Q450: 'Q451: 'Q452: 'Q453: 'Q454: 'Q455: 'Q456: 'Q457: 'Q458: 'Q459: 'Q460: 'Q461: 'Q462: 'Q463: 'Q464: 'Q465: 'Q466: 'Q467: 'Q468: 'Q469: 'Q470: 'Q471: 'Q472: 'Q473: 'Q474: 'Q475: 'Q476: 'Q477: 'Q478: 'Q479: 'Q480: 'Q481: 'Q482: 'Q483: 'Q484: 'Q485: 'Q486: 'Q487: 'Q488: 'Q489: 'Q490: 'Q491: 'Q492: 'Q493: 'Q494: 'Q495: 'Q496: 'Q497: 'Q498: 'Q499: 'Q500: 'Q501: 'Q502: 'Q503: 'Q504: 'Q505: 'Q506: 'Q507: 'Q508: 'Q509: 'Q510: 'Q511: 'Q512: 'Q513: 'Q514: 'Q515: 'Q516: 'Q517: 'Q518: 'Q519: 'Q520: 'Q521: 'Q522: 'Q523: 'Q524: 'Q525: 'Q526: 'Q527: 'Q528: 'Q529: 'Q530: 'Q531: 'Q532: 'Q533: 'Q534: 'Q535: 'Q536: 'Q537: 'Q538: 'Q539: 'Q540: 'Q541: 'Q542: 'Q543: 'Q544: 'Q545: 'Q546: 'Q547: 'Q548: 'Q549: 'Q550: 'Q551: 'Q552: 'Q553: 'Q554: 'Q555: 'Q556: 'Q557: 'Q558: 'Q559: 'Q560: 'Q561: 'Q562: 'Q563: 'Q564: 'Q565: 'Q566: 'Q567: 'Q568: 'Q569: 'Q570: 'Q571: 'Q572: 'Q573: 'Q574: 'Q575: 'Q576: 'Q577: 'Q578: 'Q579: 'Q580: 'Q581: 'Q582: 'Q583: 'Q584: 'Q585: 'Q586: 'Q587: 'Q588: 'Q589: 'Q590: 'Q591: 'Q592: 'Q593: 'Q594: 'Q595: 'Q596: 'Q597: 'Q598: 'Q599: 'Q600: 'Q601: 'Q602: 'Q603: 'Q604: 'Q605: 'Q606: 'Q607: 'Q608: 'Q609: 'Q610: 'Q611: 'Q612: 'Q613: 'Q614: 'Q615: 'Q616: 'Q617: 'Q618: 'Q619: 'Q620: 'Q621: 'Q622: 'Q623: 'Q624: 'Q625: 'Q626: 'Q627: 'Q628: 'Q629: 'Q630: 'Q631: 'Q632: 'Q633: 'Q634: 'Q635: 'Q636: 'Q637: 'Q638: 'Q639: 'Q640: 'Q641: 'Q642: 'Q643: 'Q644: 'Q645: 'Q646: 'Q647: 'Q648: 'Q649: 'Q650: 'Q651: 'Q652: 'Q653: 'Q654: 'Q655: 'Q656: 'Q657: 'Q658: 'Q659: 'Q660: 'Q661: 'Q662: 'Q663: 'Q664: 'Q665: 'Q666: 'Q667: 'Q668: 'Q669: 'Q670: 'Q671: 'Q672: 'Q673: 'Q674: 'Q675: 'Q676: 'Q677: 'Q678: 'Q679: 'Q680: 'Q681: 'Q682: 'Q683: 'Q684: 'Q685: 'Q686: 'Q687: 'Q688: 'Q689: 'Q690: 'Q691: 'Q692: 'Q693: 'Q694: 'Q695: 'Q696: 'Q697: 'Q698: 'Q699: 'Q700: 'Q701: 'Q702: 'Q703: 'Q704: 'Q705: 'Q706: 'Q7

Q11: ?

Q12: 7

Q13: 7
Q14: 6

Q14: ?
Q15: ?

Q15: 5
Q16: 6

Q16: 1

Q17: 1

Q17:
Total:

Total:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```

      Ld-(8,rosa)
    Ld-(201,tomate)
      e-(18,manjericao)
        e-(15,manjericao)
          d-(101,cravo)
        d-(301,tomate)

```

Elemento buscado: (201,tomate,astrix,2010,5)

Erro na busca: elemento não encontrado!

Levantamento do cultivar rosa: 0

```

(15,manjericao)
  e-(5,tomate)
    e-(4,tomate)
      d-(10,rosa)
    d-(101,cravo)
      e-(18,manjericao)
        d-(201,tomate)
          Ld-(303,rosa)

```

Levantamento do cultivar rosa: 5

Exemplo de Entrada 3:

```

e
l cenoura isla
b 13
f

```

Exemplo de Saída 3:

```

*arvore vazia*
Levantamento do cultivar cenoura: 0
Erro na busca: elemento não encontrado!

```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 12: Estrutura de dados AVL recursiva - Festival de Cinema

Tubiacanga irá sediar o Festival de Cinema Brasileiro no ano de 2022 e os responsáveis pelo festival decidiram fazer uma série de levantamentos de dados para, a partir disso, organizar um belíssimo festival. Para iniciar o levantamento, todos os festivais já realizados serão cadastrados com o nome da cidade em que a edição ocorreu, o ano e a quantidade de filmes que foram apresentados. Utilizando o [código fornecido](#) implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função

transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por um **ano** do tipo *inteiro*, o **nome da cidade em que o evento foi realizado** do tipo *string* e a **quantidade de filmes apresentados** do tipo *inteiro*. Não existe mais de uma edição em um dado ano.

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Entradas:

1. i - para **inserir** elemento : deve ler ano [inteiro] , nome da cidade [string] e a quantidade de filmes apresentados [inteiro]
2. r - para **remover** elemento : deve ler ano [inteiro]
3. b - para **buscar** elemento : deve ler ano [inteiro]
4. l - para fazer o **levantamento** do ano com a maior quantidade de filmes apresentados
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([ano],[nome],[quantidade])", em que [ano], [nome] e [quantidade] devem ser substituídos pelo respectivo ano, nome da cidade em que o evento ocorreu e quantidade de filmes apresentados
4. l - levantamento por ano: Imprime: "Edição com maior quantidade de filmes apresentados: [ano]", em que [ano] deve ser substituído pelo ano com a maior quantidade filmes apresentados -- no caso de empate, deve ser impresso o mais antigo
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 1992 Remanso 5
i 2011 Preciosa 12
i 2007 Passaperto 2
i 2002 Maramores 7
i 2005 Roseiral 3
i 1984 Sucupira 6
i 1989 Santana_do_Agrete 12
i 1998 Santana_dos_Ferros 15
i 1985 Asa_Branca 11
i 1988 Sensilvania 13
e
l
f
```

Exemplo de Saída:

```
(2002,Maramores)
├e-(1989,Santana_do_Agrete)
│   └e-(1985,Asa_Branca)
│       └e-(1984,Sucupira)
│           └d-(1988,Sensilvania)
├d-(1992,Remanso)
│   └d-(1998,Santana_dos_Ferros)
└d-(2007,Passaperto)
    └e-(2005,Roseiral)
```

Edição com maior quantidade de filmes apresentados: 1998

Minutos Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
i 1992 Remanso 5
i 2011 Preciosa 12
i 2007 Passaperto 2
i 2002 Maramores 7
i 2005 Roseiral 3
i 1984 Sucupira 6
i 1989 Santana_do_Agreste 12
i 1998 Santana_dos_Ferros 15
i 1985 Asa_Branca 11
i 1988 Sensilvania 13
e
r 2005
r 1989
e
f
```

```
(2002,Maramores)
|e-(1989,Santana_do_Agreste)
|   |e-(1985,Asa_Branca)
|   |   |e-(1984,Sucupira)
|   |   |d-(1988,Sensilvania)
|   |d-(1992,Remanso)
|   |   |d-(1998,Santana_dos_Ferros)
|d-(2007,Passaperto)
|   |e-(2005,Roseiral)
|   |d-(2011,Preciosa)
(2002,Maramores)
|e-(1992,Remanso)
|   |e-(1985,Asa_Branca)
|   |   |e-(1984,Sucupira)
|   |   |d-(1988,Sensilvania)
|   |d-(1998,Santana_dos_Ferros)
|d-(2007,Passaperto)
|   |d-(2011,Preciosa)
```

Exemplo de Entrada:

```
i 1992 Remanso 5
i 2011 Preciosa 12
i 2007 Passaperto 2
i 2002 Maramores 7
i 2005 Roseiral 3
i 1984 Sucupira 6
i 1989 Santana_do_Agreste 12
i 1998 Santana_dos_Ferros 15
i 1985 Asa_Branca 11
i 1988 Sensilvania 13
e
b 1984
b 2000
l
r 1992
r 2002
e
f
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Exemplo de Saída:

```
(2002,Maramores)
├─(1989,Santana_do_Agrete)
│   └─(1985,Asa_Branca)
│       └─(1984,Sucupira)
│           └─(1988,Sensilvania)
│               └─(1992,Remanso)
│                   └─(1998,Santana_dos_Ferros)
└─(2007,Passaperto)
    └─(2005,Roseiral)
        └─(2011,Preciosa)
Elemento buscado: (1984,Sucupira,6)
Erro na busca: elemento não encontrado!
Edição com maior quantidade de filmes apresentados: 1998
(2005,Roseiral)
├─(1989,Santana_do_Agrete)
│   └─(1985,Asa_Branca)
│       └─(1984,Sucupira)
│           └─(1988,Sensilvania)
│               └─(1998,Santana_dos_Ferros)
└─(2007,Passaperto)
    └─(2011,Preciosa)
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 13: Estrutura de dados AVL recursiva - Ferramentas

Uma loja de ferramentas decidiu mudar sua estratégia de vendas e, sempre que um cliente pergunta por uma ferramenta, o vendedor oferece a de maior potência que tiver na loja. As ferramentas estão cadastradas com uma chave, o nome da ferramenta, a marca e a potência. Utilizando o [código fornecido](#), implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **nome da ferramenta** do tipo *string*, a **marca** do tipo

string e a **potência** do tipo *inteiro*. Não existem ferramentas diferentes com a mesma chave.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , nome da ferramenta [string], marca [string] e a potência [inteiro]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** da marca com maior potência de uma dada ferramenta : deve ler nome da ferramenta [string]
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Informações sobre a ferramenta: ([chave],[nome],[marca],[potencia])", em que [chave], [nome], [marca] e [potencia] devem ser substituídos pela respectiva chave, nome da ferramenta, marca da ferramenta e potência da ferramenta
4. l - levantamento de ferramenta: Imprime: "Levantamento do ferramenta [ferramenta]: [marca]", em que [ferramenta] é dada como entrada para o levantamento e [marca] deve ser substituída pela marca da ferramenta mais potente -- no caso de haver mais de uma marca com a mesma potência, o resultado deverá mostrar a marca da ferramenta de menor chave
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 5 tupia Makita 650
i 7 serra_circular DeWalt 200
i 9 lixadeira Vonder 275
i 6 tupia Fort 1200
i 11 lixadeira Skill 250
i 42 serra_sabre Makita 850
i 43 serra_circular Bosch 200
i 44 furadeira Fort 650
i 45 esmerilhadeira Black&Decker 650
i 46 furadeira Schulz 500
i 101 martelete DeWalt 500
i 103 serra_circular Fort 200
e
b 45
l lixadeira
l serra_circular
f
```

Exemplo de Saída:

```
(43,serra_circular)
├─(7,serra_circular)
│   └─(5,tupia)
│       └─(6,tupia)
│           └─(11,lixadeira)
│               └─(9,lixadeira)
│                   └─(42,serra_sabre)
└─(45,esmerilhadeira)
    └─(44,furadeira)
        └─(101,martelete)
            └─(46,furadeira)
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```

      Ld-(103,serra_circular)

```

Elemento buscado: (45,esmerilhadeira,Black&Decker,650)

Levantamento do ferramenta lixadeira: Vonder

Levantamento do ferramenta serra_circular: DeWalt

Exemplo de Entrada 2:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```

i 5 tupia Makita 650
i 7 serra_circular DeWalt 200
i 9 lixadeira Vonder 275
i 6 tupia Fort 1200
i 11 lixadeira Skill 250
i 42 serra_sabre Makita 850
i 43 serra_circular Bosch 200
i 44 furadeira Fort 650
i 45 esmerilhadeira Black&Decker 650
i 46 furadeira Schulz 500
i 101 martelete DeWalt 500
i 103 serra_circular Fort 200
e
b 45
r 45
b 45
r 43
r 7
i 303 martelete Fort 600
i 404 martelete Schulz 500
i 505 martelete Makita 550
e
l lixadeira
l serra_circular
l martelete
f

```

Exemplo de Saída 2:

```

(43,serra_circular)
  e-(7,serra_circular)
    e-(5,tupia)
      Ld-(6,tupia)
    d-(11,lixadeira)
      e-(9,lixadeira)
        d-(42,serra_sabre)
    d-(45,esmerilhadeira)
      e-(44,furadeira)
        d-(101,martelete)
          e-(46,furadeira)
            d-(103,serra_circular)

```

Elemento buscado: (45,esmerilhadeira,Black&Decker,650)
 Erro na busca: elemento não encontrado!

```

(44,furadeira)
  e-(9,lixadeira)
    e-(5,tupia)
      Ld-(6,tupia)
    d-(11,lixadeira)
      d-(42,serra_sabre)
  d-(303,martelete)
    e-(101,martelete)
      e-(46,furadeira)
        d-(103,serra_circular)
    d-(404,martelete)
      d-(505,martelete)

```

Levantamento do ferramenta lixadeira: Vonder

Levantamento do ferramenta serra_circular: Fort
 Levantamento do ferramenta martetele: Fort

Exemplo de Entrada 3:

```
i 5 tupia Makita 1200
i 7 serra_circular DeWalt 200
i 9 lixadeira Vonder 250
i 6 tupia Fort 1200
i 11 lixadeira Skill 250
i 42 serra_sabre Makita 850
i 43 serra_circular Bosch 250
i 44 furadeira Fort 650
l lixadeira
l tupia
e
f
```

Exemplo de Saída 3:

```
Levantamento do ferramenta lixadeira: Vonder
Levantamento do ferramenta tupia: Makita
(7,serra_circular)
  |e-(5,tupia)
  |  |d-(6,tupia)
  |d-(11,lixadeira)
  |  |e-(9,lixadeira)
  |  |d-(43,serra_circular)
  |  |  |e-(42,serra_sabre)
  |  |  |d-(44,furadeira)
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo escolhido

Questão 14: Estrutura de dados AVL recursiva - Pokemon

Está chegando o próximo mega-híper-super campeonato de Pokemon Come e você resolve colocar à prova os seus fabulosos conhecimentos. Para isso, você resolveu fazer um programinha para cadastrar os seus pokemons e, para maior eficiência, você optou pelo uso de uma árvore binária de busca balanceada - AVL. Utilizando o [código fornecido](#), implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar as assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por um **id** do tipo *inteiro*, o **nome do pokemon** do tipo *string*, o **tipo principal do pokemon** do tipo *string* e o **nível alcançado** do tipo *inteiro*.

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Entradas:

1. i - para **inserir** elemento : deve ler id [inteiro] , nome do pokemon [string], tipo [string] e nível [inteiro]
2. r - para **remover** elemento : deve ler id [inteiro]
3. b - para **buscar** elemento : deve ler id [inteiro]
4. l - para fazer o **levantamento** de pokemons de um determinado tipo e nível: deve ler uma string e um inteiro
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([id],[nome],[tipo],[nível])", em que [id], [nome], [tipo] e [nível] devem ser substituídos pelo respectivo id, nome do pokemon, tipo principal e nível alcançado."
4. l - levantamento por marca: Imprime: "Levantamento de pokemons tipo [tipo] e nível [nível]: [quantidade]", em que [tipo] é o tipo de pokemon, [nível] o nível alcançado e [quantidade] é a quantidade de pokemons atendendo esses dois requisitos.
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada 1:

```
i 12 pikachu eletrico 25
i 51 charizard fogo 30
i 302 bulbassauo agua 40
i 14 eevee normal 45
i 96 magmar fogo 25
i 301 flareon fogo 30
i 13 jolteon eletrico 25
i 55 raichu eletrico 30
l fogo 30
l eletrico 25
l fogo 25
e
f
```

Exemplo de Saída1 :

```
Levantamento de pokemons tipo fogo e nível 30: 2
Levantamento de pokemons tipo eletrico e nível 25: 2
Levantamento de pokemons tipo fogo e nível 25: 1
(51,charizard)
├─(13,jolteon)
│   └─(12,pikachu)
│       └─(14,eevee)
└─(301,flareon)
    └─(96,magmar)
        └─(55,raichu)
            └─(302,bulbassauo)
```

Exemplo de Entrada 2:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
i 12 pikachu eletrico 25
i 51 charizard fogo 30
i 302 bulbassaur agua 40
i 14 eevee normal 45
i 96 magmar fogo 25
i 102 cleffairy normal 45
i 402 snorlax normal 45
i 9 rattata normal 45
i 77 ditto normal 42
i 66 meowth normal 23
i 88 ponyta fogo 30
i 2 charmander fogo 30
i 301 flareon fogo 30
i 13 jolteon eletrico 25
i 55 raichu eletrico 30
e
l fogo 30
r 9
r 96
r 402
b 14
b 9
l normal 45
l fogo 25
e
f
```

Exemplo de Saída 2:

```
(77,ditto)
├─e-(12,pikachu)
│   └─e-(9,rattata)
│       └─e-(2,charmander)
│           └─d-(51,charizard)
│               └─e-(14,eevee)
│                   └─e-(13,jolteon)
│                       └─d-(66,meowth)
│                           └─e-(55,raichu)
└─d-(102,cleffairy)
    └─e-(96,magmar)
        └─e-(88,ponyta)
            └─d-(302,bulbassaur)
                └─e-(301,flareon)
                    └─d-(402,snorlax)
```

Levantamento de pokemons tipo fogo e nível 30: 4

Elemento buscado: (14,eevee,normal,45)

Erro na busca: elemento não encontrado!

Levantamento de pokemons tipo normal e nível 45: 2

Levantamento de pokemons tipo fogo e nível 25: 0

```
(77,ditto)
├─e-(51,charizard)
│   └─e-(12,pikachu)
│       └─e-(2,charmander)
│           └─d-(14,eevee)
│               └─e-(13,jolteon)
└─d-(66,meowth)
    └─e-(55,raichu)
└─d-(102,cleffairy)
    └─e-(88,ponyta)
        └─d-(302,bulbassaur)
            └─e-(301,flareon)
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo Nenhum arquivo escolhido

Enviar Resposta

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Questão 15: Estrutura de dados AVL recursiva - PeguePague

Um estabelecimento comercial decidiu disponibilizar diversos pontos de consulta para que os próprios clientes consultem a disponibilidade de um produto, sirvam-se da quantidade desejada e dirijam-se ao caixa para o pagamento. Para que essa estratégia seja bem sucedida, ele o contratou para desenvolver o sistema de busca ao produtos, e, para maior eficiência, você optou pelo uso de uma árvore binária de busca balanceada - AVL. Utilizando o [código fornecido](#) implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **nome do produto** do tipo *string*, a **marca do produto** do tipo *string* e a **quantidade disponível** do tipo *inteiro*.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , nome do produto [string], marca [string] e quantidade disponível [inteiro]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** da quantidade de produtos de uma dada marca: deve ler uma string
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([chave],[nome],[marca],[quantidade])", em que [chave], [nome], [local] e [quantidade] devem ser substituídos pela respectiva chave, nome do produto, marca e quantidade disponível."
4. l - levantamento por marca: Imprime: "Levantamento da marca [marca]: [quantidade]", em que [marca] é a marca pesquisada e [quantidade] é a quantidade de produtos dessa marca.
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree

6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 100 chocolate SrCacau 30
i 187 sal MarSalgado 15
i 210 sorvete LeiGelato 28
i 130 biscoito CreckCrack 111
i 304 azeite MarSalgado 21
i 214 capuccino SrCacau 12
i 402 sardinha MarSalgado 21
l MarSalgado
l SrCacau
e
f
```

Exemplo de Saída:

```
Levantamento da marca MarSalgado: 57
Levantamento da marca SrCacau: 42
(187,sal)
├e-(100,chocolate)
├└d-(130,biscoito)
├d-(214,capuccino)
├└e-(210,sorvete)
├└└d-(304,azeite)
├└└└d-(402,sardinha)
```

Exemplo de Entrada 2:

```
i 100 chocolate SrCacau 30
i 187 sal MarSalgado 15
i 210 sorvete LeiGelato 28
i 130 biscoito CreckCrack 111
i 187 sal MarSalgado 15
i 304 azeite MarSalgado 21
i 214 capuccino SrCacau 12
i 402 sardinha MarSalgado 21
i 87 pacoquinha DaDinda 13
i 55 azeitonas Filistao 24
i 13 macarrao Filistao 12
i 803 lasanha Filistao 21
i 255 mocaccinho SrCacau 13
b 210
r 210
r 210
b 210
l SrCacau
l Filistao
e
f
```

Exemplo de Saída 2:

```
Erro na inserção: chave já existente!
Elemento buscado: (210,sorvete,LeiGelato,28)
Erro na remoção: chave não encontrada!
Erro na busca: elemento não encontrado!
Levantamento da marca SrCacau: 55
Levantamento da marca Filistao: 57
(187,sal)
├e-(100,chocolate)
├└e-(55,azeitonas)
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```

    |
    |   e-(13,macarrao)
    |   |   d-(87,pacoquinha)
    |   |   |   d-(130,biscoito)
    |   |   |   |   d-(304,azeite)
    |   |   |   |   |   e-(214,capuccino)
    |   |   |   |   |   |   d-(255,mocaccinho)
    |   |   |   |   |   |   |   d-(402,sardinha)
    |   |   |   |   |   |   |   |   d-(803,lasanha)
  
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 16: Estrutura de dados AVL recursiva - Checagem CPF

Uma empresa está tentando se adaptar a rotina de distanciamento social vigente. Uma das medidas é restringir a entrada de funcionários ao espaço físico da empresa de acordo com o número do CPF. Em dias pares, somente funcionários com CPF finalizando em número par pode entrar, em casos de dias ímpares, somente funcionários com CPF finalizando em número ímpar. Para agilizar o controle de acesso, você deve criar um sistema para fazer buscas rápidas na base, para isso, utilizando árvores binárias de buscas balanceadas.

Utilizando o [código fornecido](#), implemente uma árvore AVL com as funções de inserir, remover e buscar elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos, em casos que precise de substituição, é a substituição pelo elemento posterior (sucessor). Além disso, você deve desenvolver uma função, que, ao receber a chave de acesso do funcionário e o dia do mês, retorna se o funcionário está autorizado a entrar ou não. Outra função necessária é listar todos os funcionários que podem acessar a empresa, para isso, a função recebe o dia do mês como entrada, e realiza a impressão **em ordem** na AVL.

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido.

O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, um **nome** do tipo *string*, e um **CPF** do tipo *unsigned long long int*.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] e valor [string]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

4. v - para **verificar** funcionario : deve ler chave [inteiro] e dia do mês [inteiro]
5. l - para **listar** funcionários : deve ler dia do mês [inteiro]
6. e - para **imprimir** árvore
7. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Nome: [nome] | CPF: [cpf] | Cod: [chave] ", em que [nome], [cpf] e [chave] são as informações armazenadas do funcionário.". Caso o elemento não seja encontrado, imprime a mensagem: "Erro na busca: elemento não encontrado!".
4. v - verificar funcionario : imprime a frase: "Autorizado.", caso a paridade seja a mesma entre o CPF e o dia do mês, caso contrário imprime "Não autorizado.", ou ainda, imprime "Erro na verificacao: chave inexistente!", caso a chave informada não esteja cadastrada.
5. l - listar funcionários : deve listar os funcionários permitidos a trabalhar no dia informado, um por linha, segundo a formatação "Nome: [nome] | CPF: [cpf] | Cod: [chave] ", em que [nome], [cpf] e [chave] são as informações armazenadas do funcionário.
6. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
7. f - finalizar : finaliza o programa

Exemplo de Entrada 1:

```
i 1 Gelo_Tancredo 1111111111
i 2 Sauna_Jose 2222222222
i 3 Gray_Fernando 3333333333
i 4 Liar_Itamar 4444444444
i 5 Rico_Fernando 5555555555
i 6 Hino_Luiz 6666666666
i 7 Huck_Dilma 7777777777
i 8 Coragem_Michel 8888888888
i 9 Naro_Jair 9999999999
e
f
```

Exemplo de Saída 1:

```
(4,Liar_Itamar)
|e-(2,Sauna_Jose)
|   |e-(1,Gelo_Tancredo)
|   |d-(3,Gray_Fernando)
|d-(6,Hino_Luiz)
|   |e-(5,Rico_Fernando)
|   |d-(8,Coragem_Michel)
|       |e-(7,Huck_Dilma)
|       |d-(9,Naro_Jair)
```

Exemplo de Entrada 2:

```
i 1 Gelo_Tancredo 1111111111
i 2 Sauna_Jose 2222222222
i 3 Gray_Fernando 3333333333
i 4 Liar_Itamar 4444444444
i 5 Rico_Fernando 5555555555
i 6 Hino_Luiz 6666666666
i 7 Huck_Dilma 7777777777
i 8 Coragem_Michel 8888888888
i 9 Naro_Jair 9999999999
```

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

v 3 21
v 3 12
v 10 12
l 11
b 5
r 5
b 5
l 22
v 5 12
r 9
f

Exemplo de Saída 2:

```
Autorizado.
Não autorizado.
Erro na verificacao: chave inexistente!
Nome: Gelo_Tancredo | CPF: 11111111111 | Cod: 1
Nome: Gray_Fernando | CPF: 33333333333 | Cod: 3
Nome: Rico_Fernando | CPF: 55555555555 | Cod: 5
Nome: Huck_Dilma | CPF: 77777777777 | Cod: 7
Nome: Naro_Jair | CPF: 99999999999 | Cod: 9
Nome: Rico_Fernando | CPF: 55555555555 | Cod: 5
Erro na busca: elemento não encontrado!
Nome: Sauna_Jose | CPF: 22222222222 | Cod: 2
Nome: Liar_Itamar | CPF: 44444444444 | Cod: 4
Nome: Hino_Luiz | CPF: 66666666666 | Cod: 6
Nome: Coragem_Michel | CPF: 88888888888 | Cod: 8
Erro na verificacao: chave inexistente!
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta

Questão 17: Estrutura de dados AVL recursiva - Biblioteca

Uma biblioteca comunitária decidiu reorganizar seus livros e renovar seu sistema de consulta, facilitando a busca por títulos. Para isso, cada livro será registrado com seu nome, localização e quantidade de itens disponíveis para empréstimo. A cada empréstimo ou devolução, a quantidade de itens disponíveis para empréstimo é atualizada. Utilizando o [código fornecido](#), implemente um árvore AVL com as funções de inserir e remover elementos. A ideia principal do problema é que implementação da remoção seja utilizando recursão, não sendo necessário o uso da função transplanta. A estratégia de remoção de elementos que não estejam em nós folhas é a substituição pelo elemento posterior (sucessor).

O código fornecido já tem o menu de entrada e o método de impressão. **Não** altere essas funções. O caminho mais fácil para a solução do exercício é não

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

alterar a assinaturas das funções, dessa forma, não resultará em conflito com o código fornecido. O dado a ser armazenado na árvore é composto por uma **chave** do tipo *inteiro*, o **nome do livro** do tipo *string*, a **localização** do tipo *string* e a **quantidade de itens** disponíveis do tipo *inteiro*. Não existem livros diferentes com a mesma chave.

Entradas:

1. i - para **inserir** elemento : deve ler chave [inteiro] , nome do livro [string], localização [string] e quantidade disponível [inteiro]
2. r - para **remover** elemento : deve ler chave [inteiro]
3. b - para **buscar** elemento : deve ler chave [inteiro]
4. l - para fazer o **levantamento** de livros em uma dada localização: deve ler uma string
5. e - para **imprimir** árvore
6. f - finalizar

Saídas:

1. i - inserir elemento : A operação de inserção **não** produz saída
2. r - remover elemento : A operação de remoção **não** produz saída
3. b - buscar elemento : A operação de busca escreve a frase: "Elemento buscado: ([chave],[nome],[local],[quantidade])", em que [chave], [nome], [local] e [quantidade] devem ser substituídos pela respectiva chave, nome do livro, localização e quantidade disponível."
4. l - levantamento por localização: Imprime: "Levantamento do local [local]: [quantidade]", em que [local] é o local pesquisado e [quantidade] é a quantidade de elementos na dada localização.
5. e - imprimir árvore : imprime a árvore seguindo o padrão de formatação tree
6. f - finalizar : finaliza o programa

Exemplo de Entrada:

```
i 5 Memorias_Postumas_de_Bras_Cubas C6E01 3
i 7 Capitaes_de_Areia C4E02 5
i 9 Luciola C6E05 2
i 6 Dom_Casmurro C3E01 4
i 11 Quincas_Borba C3E01 3
i 42 O_Guia_do_Mochileiro_das_Galaxias C7E42 10
i 43 O_Restaurante_no_Fim_do_Universo C7E42 8
i 44 A_Vida_o_Universo_e_Tudo_Mais C7E42 7
i 45 Ate_mais_e_Obrigado_pelos_Peixes C7E42 6
i 46 Praticamente_Inofensiva C7E42 6
e
l C7E42
l C3E01
f
```

Exemplo de Saída:

```
(7,Capitaes_de_Areia)
├─(5,Memorias_Postumas_de_Bras_Cubas)
│   └─(6,Dom_Casmurro)
└─(43,O_Restaurante_no_Fim_do_Universo)
    ├─(11,Quincas_Borba)
    │   └─(9,Luciola)
    │       └─(42,O_Guia_do_Mochileiro_das_Galaxias)
    └─(45,Ate_mais_e_Obrigado_pelos_Peixes)
        ├─(44,A_Vida_o_Universo_e_Tudo_Mais)
        └─(46,Praticamente_Inofensiva)
Levantamento do local C7E42: 37
Levantamento do local C3E01: 7
```

Exemplo de Entrada 2:

Minutos
Restantes:
?

Usuário:
Lucas Gomes
Colombo

Notas:

Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Q7: ?
Q8: ?
Q9: ?
Q10: ?
Q11: ?
Q12: ?
Q13: ?
Q14: ?
Q15: ?
Q16: ?
Q17: ?
Total: 0

```
i 5 Memorias_Postumas_de_Bras_Cubas C6E01 3
i 7 Capitaes_de_Areia C4E02 5
i 9 Luciola C6E05 2
i 6 Dom_Casmurro C3E01 4
i 14 Pedro_e_o_Lobo C6E05 8
l C6E05
b 9
r 9
r 9
b 9
e
f
```

Exemplo de Saída 2:

```
Levantamento do local C6E05: 10
Elemento buscado: (9,Luciola,C6E05,2)
Erro na remoção: chave não encontrada!
Erro na busca: elemento não encontrado!
(7,Capitaes_de_Areia)
├e-(5,Memorias_Postumas_de_Bras_Cubas)
├└d-(6,Dom_Casmurro)
└d-(14,Pedro_e_o_Lobo)
```

Observação: para baixar o arquivo, clique com o botão direito do mouse, depois clique em "salvar link como" e escolha uma pasta do seu computador. Não tente abrir o código no navegador e copiar o texto. Isso resultará em mudança na codificação do texto de UTF-8 para ISO, deformando a saída formatada da árvore.

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo escolhido

Enviar Resposta



Desenvolvido por Bruno
Schneider a partir do programa
original (Algod) de Renato R. R.
de Oliveira.

