==========

SSC0603 – ED1 – Prof. Fernando Osório – ICMC / USP 2020-2
Prof. Fernando Osório – Estrutura de Dados 1 (Eng.Comp.)
Run.Codes: Trabalho Prático - TP-02 Versão TP-02 v.01
Trabalho: ARVDEC – Árvore de Decisão

==========

TP02 - Árvore de Decisão

>> Definição Geral do Trabalho: TADs Árvores Binárias (não ordenada!)
(*) Programa implementado na Linguagem de programação "C"
(**) Podem ser usadas as bibliotecas (rotinas) do TAD do livro do Backes (vai precisar ser adaptado para uma árvore não ordenada), ou mesmo, rotinas de outros autores que implementem Árvores Dinâmicas.

Este trabalho é relacionado ao uso de uma **lista não-linear encadeada dinâmica do tipo árvore binária,** e também pode incluir o uso de **funções recursivas** para manipular/percorrer as árvores binárias, tanto na inserção de nodos (busca) quando na consulta aos dados da árvore (visitar nodos).

O programa irá ler os dados de entrada (stdin/teclado), no formato descrito abaixo, e que irá permitir a construção da árvore. Os dados de entrada permitem construir a árvore CORRETAMENTE de cima para baixo, ou seja, da raiz em direção aos nodos folhas. Os dados são strings separadas por espaços em branco (linhas de texto: palavra1 palavra2 palavra3).

Exemplo de uma árvore de 3 nodos, criada a partir de uma descrição textual:

```
A1 B1 B2          Gera a Arvore Binária:           A1  (Raiz)
B1 S1 X                                           /  \
B2 X S2                                         B1     B2   (Filhos de A1: B1 e B2)
S1 X X                                         / \    / \
S2 X X                                        S1  X  X   S2    (X são Nodos Folha)
X X X                                        / \        / \
                                            X   X      X   X   (S são nodos Saída)
```

Sobre **os dados de entrada**: (vamos usar o teclado/stdin, ou seja, a entrada padrão do run.codes)

Cada linha da entrada é uma tripla de dados, separados por espaço em branco, contendo o dado do nodo (identificador). O identificador pode ser um número, ou um texto, como no exemplo ("A1"). Existem identificadores especiais, que são os identificadores do tipo "X" e que denotam um nodo vazio (sem filhos naquela posição). Portanto, um nodo do tipo "C1 X X") é um nodo folha, pois não tem nem um filho a esquerda e nem um filho a direita, sendo portanto, um nodo folha da árvore.

Cada linha da entrada deve ser interpretada como sendo uma tripla que contém um identificador (dado do nodo a ser inserido), seguido de dois outros identificadores, que são respectivamente o identificador da esquerda, e o identificador da direita. Portanto temos: palavra1(Pai) palavra2(FilhoEsq) palavra2(FilhoDir). Se os identificadores forem "X", significa que não temos um filho no respectivo lado onde está indicado o "X". A última linha do arquivo contém 3 identificadores "X" separados por espaço, indicando que não é para inserir mais nodos na árvore (fim da entrada de dados). A árvore construída a partir dos dados da entrada é GARANTIDAMENTE correta, ou seja, a descrição resulta em uma árvore bem estruturada e sem erros.

Processo de criação e inserção de nodos na árvore: Sugere-se um algoritmo RECURSIVO

- Se a árvore estiver vazia, a primeira linha de dados é usada para criar o nodo RAIZ e os seus devidos filhos.

- Se a árvore já tiver nodos, deve ser BUSCADO de forma RECURSIVA o identificador do nodo em questão a ser inserido, por exemplo, na linha 2 do exemplo acima, devemos buscar RECURSIVAMENTE na árvore o nodo "B1". Uma vez encontrado este nodo (e ele vai sempre existir, pois já terá sido inserido previamente), deve então ser colocados os 2 filhos associados a este nodo, como filhos do nó que já está inserido na árvore. No caso do exemplo da linha 2: busca por "B1", e ao encontrar B1", insere como seu filho a esquerda o nodo "C1" e indica que "B1" não possui nodos a direita (sem filhos a direita devido ao "X" – *Ponteiro nulo para a direita*).

- Repete este processo de busca e construção da árvore, até que encontre o identificador "X X X".

O trabalho TP-02 (Árvore de Decisão) deverá ser implementado usando os conceitos de modularidade e abstração vistos em aula, podendo ser feito a partir dos TADs de Árvores Binárias (Lista não linear) disponibilizados aos alunos (Adaptado do TAD ABO, por exemplo), ou usando outros TADs de Árvores Binárias disponíveis na literatura.

Uma vez construída a árvore, vamos exibir na tela algumas informações sobre a árvore.

Processo de visitar os nodos e exibir a saída do programa: Visitar => RECURSIVO

Os dados de saída da execução do programa, exibidos na tela, representam:

- A árvore sendo percorrida em modo recursivo e pré-fixado, indicando o número de filhos que este nodo tem e se o nodo tem filhos só a esquerda ou só a direita, em ambos lados (esquerda e direita), ou se é nodo folha, como apresentado no exemplo a seguir:

```
A1 2 ED
B1 1 E
S1 0 F
B2 1 D
S2 0 F
5 2
```

A saída indica portanto, o conteúdo do nodo (string, por exemplo "A1", "C1", ...), seguida de um número inteiro que indica o número de filhos (0 filhos é folha; 1, só um filho; 2 dois filhos), sempre separados por um espaço em branco. Cada nodo da árvore é exibido em uma linha da saída (com "\n" após cada linha com os dados do nodo). Quando o nodo não tem filhos, exibe a seguir a letra "F" indicando que é folha. Quando o nodo tem 2 filhos, exibe a seguir as letras "ED" indicando que tem filhos tanto na esquerda quanto na direita. Quando o noto tem 1 filho apenas, indica na letra qual dos lados que está este filho, "E" se só tem filho à esquerda e "D" se só tem filho à direita. E não custa lembrar, esta exibição é implementada de modo recursivo. Por fim, na última linha de saída na tela é escrito o número total de nodos que a árvore possui, onde no caso do exemplo acima são 5 nodos, e o número de nodos sem filhos, considerados como nodos de saída – S1 e S2 (escreve o valor 5 seguido de 2 e do "\n")

>> O trabalho deve ser entregue via RUN.CODES

>> A definição do trabalho e exemplos de casos de teste estão disponíveis na página da Wiki:

http://wiki.icmc.usp.br/index.php/TrabPrat_SSC0603_2020(fosorio)

>> Exemplos de Dados dos Casos de Teste:

Entrada:

```
A1 B1 B2
B1 S1 X
B2 X S2
S1 X X
S2 X X
X X X
```

Saída: (na tela)

```
A1 2 ED
B1 1 E
S1 0 F
B2 1 D
S2 0 F
5 2
```

SEGUNDO EXEMPLO DE DADOS

Entrada:

```
Febre Tosse Dor
Dor S-Aspirina2 S-AssitirTV
S-Aspirina2 X X
S-AssitirTV X X
Tosse S-Hospital S-Aspirina1
S-Hospital X X
S-Aspirina1 X X
X X X
```

Saída:

```
Febre 2 ED
Tosse 2 ED
S-Hospital 0 F
S-Aspirina1 0 F
Dor 2 ED
S-Aspirina2 0 F
S-AssitirTV 0 F
7 4
```

```
===========
  F.Osório
  Nov.2020
===========
```