*"""  
Henry Ang  
CSC 4800 Advanced Python  
1/26/17  
Lab 4 - RegEx: Stock Quote XML  
  
This program acesses the yahoo finance website to obtain the latest stock price information  
about one or more stock symbols.  
"""***import** urllib.request, re, sys  
  
**def** ProcessQuotes(strSyms, sym):  
 *"""  
 Process a stock quote yahoo finance website and prints out in XML format* **:param** *strSyms:* **:param** *sym: User input of stock symbol  
 """* strUrl=**'http://finance.yahoo.com/d/quotes.csv?f=sd1t1l1bawmc1vj2&e=.csv'** strUrl = strUrl + strSyms  
 **try**:  
 f = urllib.request.urlopen(strUrl)  
  
 **except**:  
 *# catch the expection if cant read url* print(**"URL access failed:\n"** + strUrl)  
 **return  
  
 for** line **in** f.readlines():  
 line = line.decode().strip(); *# convert byte array to string* print(line, **'\n'**)  
  
 **if** line == **"\""** + sym + **"\""** + **",N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A,N/A"**: *# if stock symbol is unknown* print(**"Unknown symbol: match failed\n"**)  
  
 **else**:  
 *# print the stock quote data* print(**"<stockquote>"**)  
 symbol(line)  
 date(line)  
 time(line)  
 lastSalePrice(line)  
 bidPrice(line)  
 askPrice(line)  
 weekLow(line)  
 weekHigh(line)  
 todayLow(line)  
 todayHigh(line)  
 netChangePrice(line)  
 shareVolumeQty(line)  
 totalShares(line)  
 print(**"</stockquote>\n"**)  
  
**def** symbol(line):  
 *"""  
 Prints the stock quote symbol in XML format* **:param** *line:  
 """* symbol = re.match(**"[\"][a-zA-z]+[\"]"**, line).group()  
 symbols = symbol.strip(**"\""**)  
 print(**"\t<qSymbol>"** + symbols + **"</qSymbol>\n"**, end=**''**)  
  
**def** date(line):  
 *"""  
 Prints the stock quote date in XML format* **:param** *line:  
 """* date = re.match(**"(.\*?)(\d+/\d+/\d+)"**, line)  
 **if** date **is None**:  
 **pass  
 else**:  
 other, dateFinal = date.groups()  
 print(**"\t<qDate>"** + dateFinal + **"</qDate>\n"**, end=**''**)  
  
**def** time(line):  
 *"""  
 Prints the stock quote time in XML format* **:param** *line:  
 """* time = re.match(**"(.\*?)(\d+:\d\d[pm|am]+)"**, line)  
 **if** time **is None**:  
 **pass  
 else**:  
 other, timeFinal = time.groups()  
 print(**"\t<qTime>"** + timeFinal + **"</qTime>\n"**, end=**''**)  
  
**def** lastSalePrice(line):  
 *"""  
 Prints the stock quote lastSalePrice in XML format* **:param** *line:  
 """* lastSalePrice = re.match(**"(.\*?)(\d+[.]\d+)"**, line)  
 **if** lastSalePrice **is None**:  
 **pass  
 else**:  
 other, lastSalePriceFinal = lastSalePrice.groups()  
 print(**"\t<qLastSalePrice>"** + lastSalePriceFinal + **"</qLastSalePrice>\n"**, end=**''**)  
  
**def** bidPrice(line):  
 *"""  
 Prints the stock quote bidPrice in XML format* **:param** *line:  
 """* bidPrice = re.match(**"(.\*?)(\d+[.]\d+[,])(\d+[.]\d+[,])"**, line)  
 **if** bidPrice **is None**:  
 **pass  
 else**:  
 other, lastSalesPrice, bidPrices = bidPrice.groups()  
 bidPricesFinal = bidPrices.strip(**","**)  
 print(**"\t<qBidPrice>"** + bidPricesFinal + **"</qBidPrice>\n"**, end=**''**)  
  
**def** askPrice(line):  
 *"""  
 Prints the stock quote askPrice in XML format* **:param** *line:  
 """* askPrice = re.match(**"(.\*?)(\d+[.]\d+[,])(\d+[.]\d+[,])(\d+[.]\d+[,])"**, line)  
 **if** askPrice **is None**:  
 **pass  
 else**:  
 other, lastSalesPrice, bidPrices, askPrices = askPrice.groups()  
 askingPrice = askPrices.strip(**","**)  
 print(**"\t<qAskPrice>"** + askingPrice + **"</qAskPrice>\n"**, end=**''**)  
  
**def** weekLow(line):  
 *"""  
 Prints the stock quote symbweekLow in XML format* **:param** *line:  
 """* weekLow = re.match(**"(.\*?)(\d+[.]\d+[ -])"**, line)  
 na = re.match(**"(.\*?)(\"\d+[.]\d+[ -]+\d+[.]\d+\")"**, line)  
 **if** na **is None**:  
 **pass  
 else**:  
 other, weekLows = weekLow.groups()  
 weekLowFinal = weekLows.strip(**" "**)  
 print(**"\t<q52WeekLow>"** + weekLowFinal + **"</q52WeekLow>\n"**, end=**''**)  
  
**def** weekHigh(line):  
 *"""  
 Prints the stock quote weekHigh in XML format* **:param** *line:  
 """* weekHigh = re.match(**"(.\*?)([- ]\d+[.]\d+)"**, line)  
 na = re.match(**"(.\*?)(\"\d+[.]\d+[ -]+\d+[.]\d+\")"**, line)  
 **if** na **is None**:  
 **pass  
 else**:  
 other, weekHighs = weekHigh.groups()  
 weekHighFinal = weekHighs.strip(**" "**)  
 print(**"\t<q52weekHigh>"** + weekHighFinal + **"</52WeekHigh>\n"**, end=**''**)  
  
**def** todayLow(line):  
 *"""  
 Prints the stock quote todayLow in XML format* **:param** *line:  
 """* todayLow = re.match(**"(.\*?)(.\*?\d+[.]\d+[ -])+"**, line)  
 na = re.match(**"(.\*?)(\"\d+[.]\d+[ -]+\d+[.]\d+\")([,]\"\d+[.]\d+[ -]+\d+[.]\d+\")"**, line)  
 **if** na **is None**:  
 **pass  
 else**:  
 other, todayLows = todayLow.groups()  
 a , b = todayLows.split(**","**)  
 todayLowFinal = b.strip(**" \""**)  
 print(**"\t<qTodaysLow>"** + todayLowFinal + **"</qTodaysLow>\n"**, end=**''**)  
  
**def** todayHigh(line):  
 *"""  
 Prints the stock quote todayHigh in XML format* **:param** *line:  
 """* todayHigh = re.match(**"(.\*?)(.\*?[- ]\d\d[.]\d+)+"**, line)  
 na = re.match(**"(.\*?)(\"\d+[.]\d+[ -]+\d+[.]\d+\")([,]\"\d+[.]\d+[ -]+\d+[.]\d+\")"**, line)  
 **if** na **is None**:  
 **pass  
 else**:  
 other, todayHighs = todayHigh.groups()  
 a, b = todayHighs.split(**"-"**)  
 todayHighFinal = b.strip(**" "**)  
 print(**"\t<qTodaysHigh>"** + todayHighFinal + **"</qTodaysHigh>\n"**, end=**''**)  
  
**def** netChangePrice(line):  
 *"""  
 Prints the stock quote netChangePrice in XML format* **:param** *line:  
 """* netChangePrice = re.match(**"(.\*?)([+-]\d[.]\d+)"**, line)  
 **if** netChangePrice **is None**:  
 **pass  
 else**:  
 other, netChangePrices = netChangePrice.groups()  
 print(**"\t<qNetChangePrice>"** + netChangePrices + **"</qNetChangePrice>\n"**, end=**''**)  
  
**def** shareVolumeQty(line):  
 *"""  
 Prints the stock quote shareVolumeQty in XML format* **:param** *line:  
 """* shareVolumeQty = re.match(**"(.\*?)([+-]\d[.]\d+)([,]\d+)"**, line)  
 **if** shareVolumeQty **is None**:  
 **pass  
 else**:  
 other, netChangePrice, shareVolumeQtys = shareVolumeQty.groups()  
 shareVolumeQtysFinal = shareVolumeQtys.strip(**", "**)  
 print(**"\t<qShareVolumeQty>"** + shareVolumeQtysFinal + **"</qShareVolumeQty>\n"**, end=**''**)  
  
**def** totalShares(line):  
 *"""  
 Prints the stock quote totalShares in XML format* **:param** *line:  
 """* totalShares = re.match(**"(.\*?)(.\*?\d+[, ]\*)+"**, line)  
 **if** totalShares **is None**:  
 **pass  
 else**:  
 other, totalShares = totalShares.groups()  
 todayHighFinal = totalShares.strip(**", "**)  
 print(**"\t<qTotalOutstandingSharesQty>"** + todayHighFinal + **"</qTotalOutstandingSharesQty>\n"**, end=**''**)  
  
**def** main():  
 *"""  
 Main function of the program. Ask user to input stock symbol and calls ProcessQuote  
 """* **while True**:  
 print(**"Enter a Stock Symbol: "**)  
 sym = sys.stdin.readline().strip()  
 **if** (len(sym) == 0):  
 **break** strSyms = **'&s='** + sym  
 ProcessQuotes(strSyms, sym)  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()