

Re: Making a decimal to fraction converter

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2008-10/msg00246.html>

- *From:* "Kasha" <RomulanPrince@xxxxxxxxxxxxx>
 - *Date:* Tue, 7 Oct 2008 16:43:10 -0400
-

Thanks! Works great =]

-Jason

"James Hahn" <jhahn@xxxxxxxxxx> wrote in message
<news:%23w5WwOqIJHA.1156@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

This is the full text of a program that converts a decimal number to a fraction, and handles repeating decimals correctly, including the case where there is a portion of the fractional part that is non-recurring. I have almost followed the form layout in your example, although two extra text boxes are required to demonstrate the parameters used for the recurring decimal case - you could simply make these internal variables instead of text box values.

It is set up to test using manually entered numbers, for reasons that become clear below.

For instance:
1.25 1 1 4
5.75 5 3 4
1.3 1 3 10
1.33 1 1 3
1.7272 1 8 11
0.8666 0 13 15
1.31999 1 8 25

Note that the repeating test is not foolproof - for instance it will detect the repeating part of 1.25252 as 52, which might not be what was intended. The cases where it fails are ambiguous, and I don't think there is any way around the issue without some very arbitrary decisions or asking the operator to indicate exactly what the recurring part is. This also means it will NOT work properly with decimal numbers created from a fraction, because the computer will usually round the last digit. So 1.66667 will not be detected as repeating. This is the problem with your original test procedure - you were producing decimal numbers from a fraction, but those numbers could not be expressed in the machine as a proper repeating decimal, so they would never be converted back to the same fraction, however good your conversion was.

```
Imports System.Text.RegularExpressions  
Public Class Form1
```

Re: Making a decimal to fraction converter

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Button1.Click
    Dim s() As String
    Dim value As Double, numer As Double, denom As Double
    Dim repeats As Double, startat As Double
    TxtRepeats.Text = "0"
    TxtStartAt.Text = "0"

    If InStr(txtWhole.Text, ".") = 0 Then Beep() : Exit Sub
    s = Split(txtWhole.Text, ".")
    If s(0) = "" Then Beep() : Exit Sub
    If s(1) = "" Then Beep() : Exit Sub
    CheckRepeating(s(1))

    value = Val(txtWhole.Text)
    repeats = Val(TxtRepeats.Text) 'Number of digits per repeat sequence
    startat = Val(TxtStartAt.Text) 'Position (base 1) of first digit in first repeat seq.
    startat -= 1
    If startat <= 0 Then startat = repeats

    If repeats = 0 Then
        denom = 10 ^ Len(s(1))
        numer = Val(txtWhole.Text) * denom
    Else
        Dim Q As String = s(1).Substring(0, startat)
        Dim R As String = s(1).Substring(startat)
        Dim Q1 As Integer = Val(Q & R.Substring(0, repeats)) - Val(Q)
        denom = Val(StrDup(CInt(repeats), "9") & StrDup(Q.Length, "0"))
        numer = Q1 + Val(s(0)) * denom
    End If

    txtNumer.Text = numer.ToString
    txtDenom.Text = denom.ToString
    Me.Refresh()
    Simplify()

End Sub
Sub CheckRepeating(ByVal s As String)
    Dim MyRegex As Regex = New Regex("(d+)\1{1,}\z")
    If Not MyRegex.IsMatch(s) Then Exit Sub

    Dim ms As MatchCollection = MyRegex.Matches(s)
    Dim m As String = ms.Item(ms.Count - 1).Groups("1").Value

    TxtRepeats.Text = m.Length
    TxtStartAt.Text = InStr(s, m)

End Sub

Private Sub Simplify()
    Dim n As Long = Val(txtNumer.Text)
```

Re: Making a decimal to fraction converter

```
Dim d As Long = Val(txtDenom.Text)
Dim i As Long

For i = n To 1 Step -1
If (n / i = Int(n / i)) And (d / i = Int(d / i)) Then Exit For
Next
n = n / i
d = d / i
i = 0

If d > 0 Then
While n > d
i = i + 1 : n = n - d
End While
End If
TxtResultW.Text = i.ToString
TxtResultN.Text = n.ToString
TxtResultD.Text = d.ToString

End Sub
End Class
```

"Kasha" <RomulanPrince@xxxxxxxxxxxx> wrote in message
news:036DC4C7-5214-4642-9F75-633F3D482485@xxxxxxxxxxxxxxxx

I AM using three text boxes =] , one for the whole number, one for the numerator, and one for the denominator.

```
Dim denomtest As Integer
Dim numertest As Integer
Dim dota As Boolean
Dim dotb As Boolean
Dim time As Integer = 1
Dim num As Integer = 2
Dim numertesthold As Integer
Dim denomtesthold As Integer
Dim numerteststa As Integer
Dim denomteststa As Integer
Dim frac As Double
```

```
Dim i As Integer
```

```
Private Sub convert()
```

Re: Making a decimal to fraction converter

```
IF int( numer.Text ) = 66 THEN
numertest = 1
denomtest = 6
Goto chocolate cake
Endif
```

```
time = 1
```

```
denomtesthold = denom.Text
numertesthold = numer.Text
```

```
top:
Do Until time = 100
```

```
For num = 2 To 51
If num = 51 Then GoTo chocolate
numertesta = numertesthold
denomtesta = denomtesthold
```

```
numertest = numertesta / num
denomtest = denomtesta / num
```

```
For i = 2 To Len(numertest)
If dota = True Or dota = False Then GoTo Here
If Mid(numertest, i, 1) = "." Then dota = True
If i = Len(numertest) Then dota = False
Next
```

```
Here:
```

```
For i = 2 To Len(denomtest)
If dotb = True Or dotb = False Then GoTo There
If Mid(denomtest, i, 1) = "." Then dotb = True
If i = Len(denomtest) Then dotb = False
Next
```

```
There:
```

```
If dota = False And dotb = False And denomtest <> 0 And numertest <> 0
And numertest / denomtest = frac Then
i = 2
time = time + 1
denomtesthold = denomtest
```

Re: Making a decimal to fraction converter

```
numertesthold = numertest  
GoTo top  
  
End If  
  
Next  
  
Loop  
  
chocolate:  
  
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
Whole.Text = Int(Inch.Text)  
numer.Text = (Inch.Text - whole.Text) * 10000000  
denom.Text = 10000000  
frac = numer.Text / denom.Text  
  
Call convert()  
Whole.Text = Int(Inch.Text)  
Numer.Text = numertesthold  
Denom.Text = denomtesthold  
  
End Sub  
End Class
```

That's how far I currently am. If I were to use your code, where would I place it?

"James Hahn" <jhahn@xxxxxxxx> wrote in message
news:OvoqMUcIJHA.4416@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Firstly, you can present the results from the three text boxes as a string:

```
text1.text & " " & text2.text & "/" & Text3.text
```

Secondly, any repeating decimal can be represented as the correct fraction using code, but you haven't indicated what your problem with this procedure is.

How is the user indicating that the decimal number is repeating indefinitely, or is the problem that you can't detect

Re: Making a decimal to fraction converter

when it's a repeating decimal? This could probably be solved with a regular expression.

Or is the problem that you don't know how to reduce a regular repeating number to a fraction? This is relatively easy, once you know the characteristics. For instance, if its a simple set of repeating digits, just use:

(Repeats is the number of digits that are repeated, textbox1 has the number to convert)

```
s = Split(TextBox1.Text, ".")  
s(0) = s(0) & s(1).Substring(0, repeats)
```

```
numerator = Val(s(0) & "." & s(1))
```

```
power = 10 ^ repeats
```

```
numerator = numerator * power
```

```
denominator = power - 1
```

then simplify if possible. If it's some unique digits followed by a repeating portion you need to allow for that in the adjustment of the value, and then adjust the numerator and denominator by powers of 10 until you have eliminated the fractional part of the numerator.

"Kasha" <RomulanPrince@xxxxxxxxxxxx> wrote in message
news:8350ED0C-6972-42CB-81BE-38C9509B1A6F@xxxxxxxxxxxxxxxxxxxx

The goal of the program is to convert decimals to fractions with only the input of a decimal number, and thus far, I've only been able to simplify nonrepeating decimal numbers. If I were to have the user specify the finest graduation, I'd need a drastically different code, and I'd be going beyond the goal of the program =\

Is my goal hopeless without specifying the finest graduation then?

-Jason
"Tom Dacon"
<tdacon@xxxxxxxxxxxxxxxxxxxx> wrote in message
news:ui4sIPNIJHA.3824@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Re: Making a decimal to fraction converter

Jason, if you can have your users specify the finest graduation to output (say, 64ths, 8ths, quarters, etc.), or if you assume a certain finest graduation, you can convert a repeating fraction and then round to the finest graduation that the number converts to, and then reduce it if possible.

So for instance, your $1/3$ is $0.33333\dots$. If your finest graduation is 128ths, multiply the $0.33333\dots$ by 128, getting $42.66666\dots$ and round to 43, giving you $43 / 128$ ths.

Similarly, $1/9$ is $0.11111\dots$. For 128ths, multiply $0.11111\dots$ by 128, getting $14.22222\dots$. Round to $14/128$ ths and reduce to $7/64$ ths.

Tom Dacon
Dacon Software Consulting

"Jason"

<RomulanPrince@xxxxxxxxxxxxxx>

wrote in message

news:204A45ED-4C6D-43E6-AD43-BBCCA0F71073@xxxxxxxxxxxxxxxxxx

I'm making
a program
that will
convert
decimal
inputs (in
this case, in
inches) and
output a
fractional
answer. At
the
moment,
I'm only
able to

Re: Making a decimal to fraction converter

output the fractional answer in three parts: A whole number text box, a numerator text box, and a denominator text box. I realize that this is probably the closest I'll get to having fractions displayed in VB, so that's no big deal. I'm able to simplify most numbers with some code I've written, but I'm unable to simplify anything that repeats infinitely (i.e.. $1/3$). Is there any way to be able to simplify this?

I can give you the code I've already written if that would help.

Thanks!

Re: Making a decimal to fraction converter