

Re: C Programming: A Modern Approach – Chapter 15 Exercise 5

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-07/msg02279.html

- *From:* Tim Rentsch <txr@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* 22 Jul 2005 10:56:23 -0700
-

richard@xxxxxxxxxxxxxxxx (Richard Tobin) writes:

> In article <pan.2005.07.22.09.48.30.184627@xxxxxxxxxxxx>,
> Simon Morgan <me@xxxxxxxxxxxx> wrote:
>
>>The task is to modify it so that it alternates between distribution of the
>>extra spaces favouring the end of the line and the beginning of the line.
>
> It's not really answering your question, but this is an example of the
> large class of problems that can be solved with something equivalent
> to Bresenham's algorithm. That algorithm is traditionally used for
> drawing straight lines on a pixel display, but can be used for all
> kinds of problems where you want to approximate division using integer
> increments.
>
> In this case, suppose you want to divide 8 spaces among 5 gaps. Start
> with zero. At each gap, then add 8 and subtract off as many 5s as you
> can, and for each one add a space to that gap. Then proceed to the
> next gap using the remainder as the starting value.
>
> So we have
>
> $0+8 = 8 = 1*5 + 3 \Rightarrow$ 1 space
> $3+8 = 11 = 2*5 + 1 \Rightarrow$ 2 spaces
> $1+8 = 9 = 1*5 + 4 \Rightarrow$ 1 space
> $4+8 = 12 = 2*5 + 2 \Rightarrow$ 2 spaces
> $2+8 = 10 = 2*5 \Rightarrow$ 2 spaces

Right, except Bresenham's algorithm usually adds 2x number of spaces
at each gap, and subtracts 2x number of gaps for each "increment"
(which is printing a space in this case), so that the comparison value
of 1x gaps can be used, which is "1/2" in the rationalized number
system.

Re: C Programming: A Modern Approach – Chapter 15 Exercise 5

• **References:**

◆ **[C Programming: A Modern Approach – Chapter 15 Exercise 5](#)**

◇ *From:* Simon Morgan

◆ **[Re: C Programming: A Modern Approach – Chapter 15 Exercise 5](#)**

◇ *From:* Richard Tobin

• Prev by Date: **[Re: C Programming: A Modern Approach – Chapter 15 Exercise 5](#)**

• Next by Date: **[Re: why is it so ?](#)**

• Previous by thread: **[Re: C Programming: A Modern Approach – Chapter 15 Exercise 5](#)**

• Next by thread: **[Re: C Programming: A Modern Approach – Chapter 15 Exercise 5](#)**

• Index(es):

◆ **[Date](#)**

◆ **[Thread](#)**