

## 1 Build Your Own Step Pyramid

Recall that `DrawRect(a,b,w,h,c)` draws a rectangle whose lower-left corner is at  $(a,b)$ , whose horizontal width is  $w$  and vertical height is  $h$ , in color  $c$ .

Suppose we are given  $L$  and  $H$ . The first (bottom) rectangle is easy to draw:

`DrawRect(0,0,L,H,'y')`

(you can change the color to any you want). Now, we need to stack another narrower rectangle on top of it. Eventually, we will need to call `DrawRect` again with certain parameters. First, the height remains the same at  $H$ , but the width is now  $2L/3$ . Next, let us determine the lower-left coordinate of this rectangle. Since the height is  $H$ , the  $y$ -coordinate must be  $H$  this time. What about the  $x$ -coordinate? Because we want this rectangle to center on top of the bottom one, we must leave equal space on both sides. How much space do we have? Since the rectangle below is  $L$  wide, and the one we are drawing is  $2L/3$  wide, this leaves  $L/3$  space. Therefore, we should leave  $(L/3)/2 = L/6$  on each side of this rectangle. That is, the  $x$ -coordinate of the lower-left corner is  $L/6$ .

How do we repeat for subsequent rectangles? We proceed almost identically to what's above. Suppose the rectangle just below the one we are drawing has the lower-left corner at  $(a,b)$  and width  $\ell$  (remember that its height is always  $H$ ). The  $y$ -coordinate must be  $H$  more than  $b$  this time, i.e.,  $b + H$ . What about the  $x$ -coordinate? Because we want this rectangle to center on top of the bottom one, we must leave equal space on both sides. How much space do we have? Since the rectangle below is  $\ell$  wide, and the one we are drawing is  $2\ell/3$  wide, this leaves  $\ell/3$  space. Therefore, we should leave  $(\ell/3)/2 = \ell/6$  on each side of this rectangle. That is, we shift the rectangle to the right  $\ell/6$  from where the left edge of the one below is, so the  $x$ -coordinate of the lower-left corner is  $a + \ell/6$ .

Thus, if we keep track of  $a$  and  $b$ , setting them both to 0 in the beginning, keep track of the current width  $\ell$ , and keep updating their values after drawing each rectangle, we will draw the pyramid correctly. Finally, how long do we do this? Since the top step must have a length no less than  $H$ , we should keep drawing when the width  $\ell$  is still greater than or equal to  $H$ , i.e., while  $\ell \geq H$ .

## 2 Fibonacci Numbers

- (a) This is not too difficult, but there is a catch if the programmer is not careful enough. In specifying  $\frac{1+\sqrt{5}}{2}$ , we have to put parentheses correctly. We also have to make sure to enter the correct error threshold. Last but not least, we should associate  $f_n$  and  $f_{n+1}$  with `f_old` and `f_cur` correctly, but this totally depends on which values are printed, etc. That is, this part should look like

```
abs(f_cur/f_old-(1+sqrt(5))/2)>0.000001
```

Incorrect expressions will lead to  $n = 1476$ ,  $n = 17$ , or  $n = 15$ . (The result is  $n = 16$ .) Note that the solution on the course webpage is incorrect: The logical expression for the while loop should be exactly as above, or (as another fix) is to change to `f_old=0`; instead, but do only one of these, not both. As we see, it is really easy to make a mistake in this kind of problems! Does your program divide anything by zero?

- (b) This is, again, a while-loop program. In order to print *all* the values less than one million, we just have to change the condition for the loop to limit on the *value* of the Fibonacci number instead of the argument  $n$ . (The last value printed is  $f_{30}$ .) Now, in order to avoid printing the values less than or equal to ten thousand, we just have to add in the lower bound. The first number we start printing is one just greater than 10,000. So, if the value has not exceeded 10,000 yet, do not print it out! ( $f_{21}, \dots, f_{30}$  are printed.)

### 3 Binomial Coefficients

This is the most difficult task to understand in the first place, but once we do, it becomes not so bad. Just to clarify: This exercise does *not* depend on the input  $n$ . Instead, we always run from  $n = 1$  to 10 and, for each  $n$ , print out the number of digits of  $\binom{n}{k}$  for  $k = 1, \dots, n$  on the same line. That means the output for different  $n$ 's should be on separate lines. If programmed correctly, your solution should output 55 numbers that form a triangular shape, with base at the bottom.

A question that popped up was how to print several numbers on the same line. The trick is to use `fprintf`; `disp` does not work because it always prints a new-line character at the end. We do not print the new-line character (`\n`) until we are done with a line. At that point, `fprintf('\n')` should do the job.

Finally, how do we display the number of digits of  $\binom{n}{k}$ ? Simple: First, calculate the actual value of  $\binom{n}{k}$ , and then use the formula `floor(log10(x))+1`, where we replace  $x$  with the value of  $\binom{n}{k}$ .

### 4 ASCII Drawing in the Command Window

Write a script that “draws” the figure below in the *Command Window* using `fprintf` statements. Prompt the user to input an integer  $n$  for the number of asterisks on each side of the square. Assume  $n > 3$ .

```
*****
**  *
*  *  *
*   **
*****
```

Recall that `fprintf(' ');` prints a single blank (space) while `fprintf('\n');` starts a new line.

**Solution:** Just to clarify, the figure above is for  $n = 5$ . Here comes the solution. First, observe that we need to print at all columns in the first and last rows, and the first and last columns always need

to be printed. The more difficult part is the diagonal inside. But this is not that hard: The second row is at the second position, the third row is at the third position, and so on. In general, for the  $i^{\text{th}}$  row, where  $2 \leq i \leq n - 1$ , the asterisk should be printed at the  $i^{\text{th}}$  column. Lastly, do not forget to print a new line after a line is completely printed.