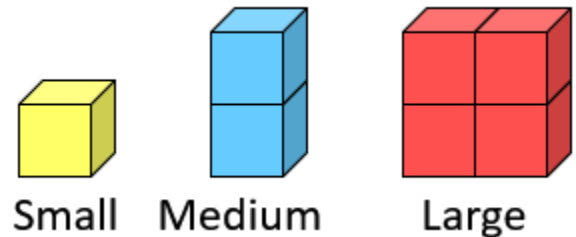


COMP2401 - Assignment #1

(Due: Monday, September 28th, 2020@ 6pm)

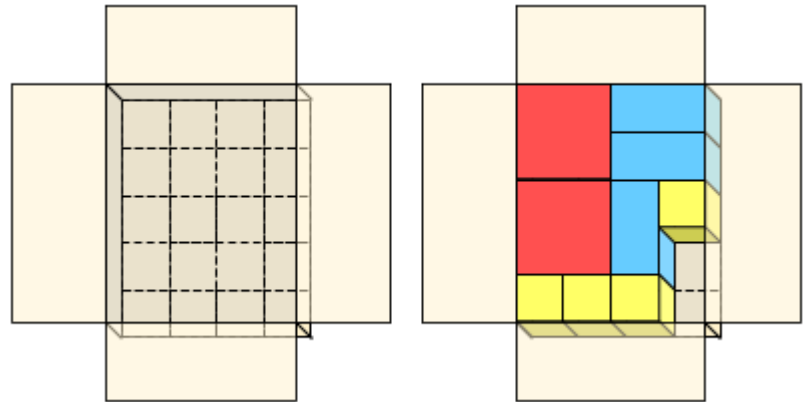
In this assignment, you will get used to working with the C language, the Linux/VirtualBox environment. The program will require you to use IF statements, FOR loops, constants, variables, arrays and random numbers.

Consider a company that sells a single product that comes either individually, in a 2-pack or in a 4-pack which are shrink-wrapped into three sizes of packages as shown here. The single package is considered to be 1x1. The medium package may be a 1x2 or 2x1 package, depending on if it is vertical or horizontal. The large is a 2x2 package.



These packages are shipped in cardboard boxes (see picture here) that can hold exactly **20** of the small-sized packages ... 4 single packages across and 5 single packages down. No packages can be stacked on top of one another.

The different sized packages will be packed in the box (as shown in the picture here on the far right) as tightly as possible.



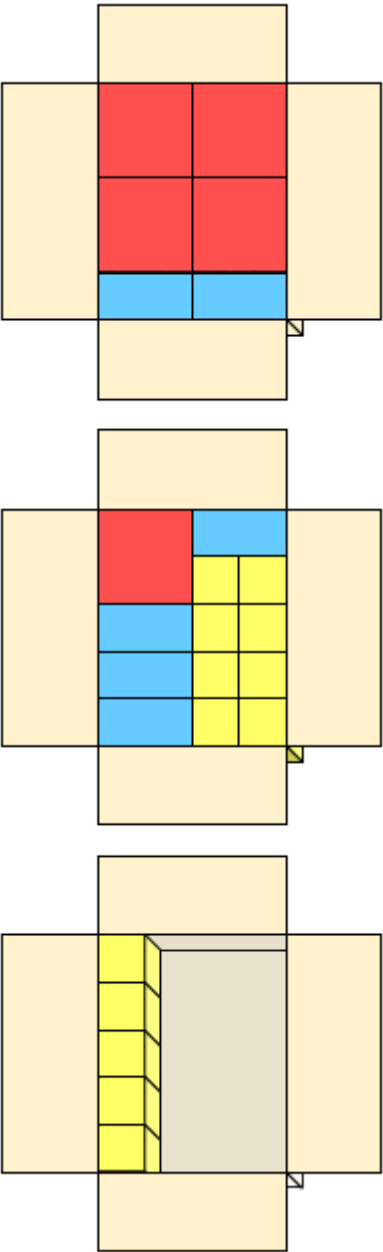
Write a program called **packBoxes.c** that generates **24** packages at random. That is, you should generate a random number of large packages, a random number of medium packages and a random number of small packages such that the total number of packages is always 24. The code should generate these numbers differently each time it is run.

Your goal is to then pack the packages into the cardboard boxes so that it minimizes the number of boxes used. Note that the medium packages might better fit vertically or horizontally ... try both. Your code should assume that at most **6** boxes are available. You should display the number of packages of each size and then the final packing of the boxes which indicates, using text, where the large, medium and small packages are in the box. One way to do this is to use letters L, M and S. For the above packing ... it would perhaps look as shown here on the right. Note that **the packing may differ in your output ... so you do not need to produce exactly what is shown here.**

LLMM
LLMM
LLMS
LLM
SSS

You may use any arrays or variables that you would like. You should make your code as efficient as possible and easily modifiable if the parameters change. For example, if you decide to change the maximum number of cardboard boxes available, the number of packages to be generated or the dimensions of each cardboard box ... then you should only have to make this change in one spot in your program. Assume, however, that the package sizes are 1x1, 1x2 (or 2x1) and 2x2 permanently ... so these can be hard-coded anywhere.

You will need to display the order information, a breakdown of the costs (i.e., number of units ordered for each package size and their costs). Assume \$5.99 for small, \$10.99 for medium and \$19.99 for large. Show the total, shipping costs (i.e., \$1.49 per cardboard box used), HST and the amount charged. Then, show how each individual box is packed. Do NOT show packing of empty boxes. Here below (on the left side) is a sample output and what it would look like in real life. Make sure that things are nicely presented and lined up well. Make sure to test your code thoroughly.

<pre> student@COMP2401-F20:~\$./packBoxes Order: Large=5, Medium=6, Small=13 Small (13 units) \$ 77.87 Medium (06 units) \$ 65.94 Large (05 units) \$ 99.95 ----- Total \$243.76 Shipping (3 boxes) \$ 4.47 HST (13%) \$ 31.69 ----- Amount Charged \$279.92 Individual boxes are packed as follows: LLLL LLLL LLLL LLLL MMMM LLMM LLSS MMSS MMSS MMSS S S S S S student@COMP2401-F20:~\$ </pre>	
--	--

See next page for submission instructions ...

IMPORTANT SUBMISSION INSTRUCTIONS:

Submit all of your **c source code** files (in this case there is just one) as a single **tar** file containing:

1. A **Readme** text file containing
 - your name and studentNumber
 - a list of source files submitted
 - any specific instructions for compiling and/or running your code
2. All of your **.c source** files and all other files needed for testing/running your programs.
3. Any output files required, if there are any.

The code **MUST** compile and run on the course VM.

- If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !
 - You WILL lose marks on this assignment if any of your files are missing. So, make sure that you hand in the correct files and version of your assignment. You will also lose marks if your code is not **written neatly with proper indentation and containing a reasonable number of comments**. See course notes for examples of what is proper indentation, writing style and reasonable commenting).
-