

Assignment 1, Part A: Billboard

(20%, due 5:00pm Wednesday, September 13th)

Overview

This is the first part of a two-part assignment. This part is worth 20% of your final grade for IFB104. Part B will be worth a further 5%. Part B is intended as a last-minute extension to the assignment, thereby testing the maintainability of your code, and the instructions for completing it will not be released until Week 8. Whether or not you complete Part B you will submit only one file, and receive only one assessment, for the whole 25% assignment.

Motivation

One of the most basic functions of any IT system is to process a given data set to produce some form of human-readable output. This assignment requires you to produce a visual image by following instructions stored in a list. It tests your abilities to:

- Process lists of data values;
- Design a solution to a computational problem;
- Display information in a visual form; and
- Produce maintainable, reusable code.

Goal

Advertising billboards are a familiar, if annoying, feature of the urban landscape. Today many such billboards are electronic, although there are still plenty of traditional ones consisting of several rectangular sheets of paper pasted onto a wooden backing board.

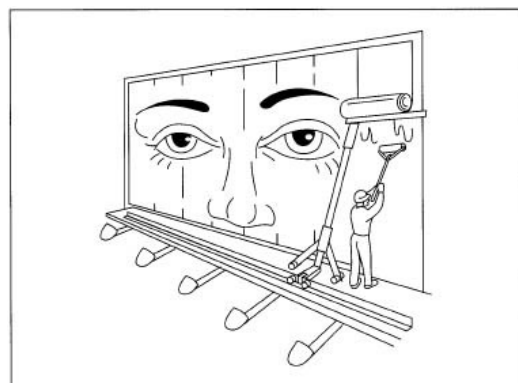


In this assignment you are required to develop a Python 3 program which processes data stored in a list to display a specific arrangement of sheets on a billboard. Each sheet must be a distinct part of a single overall image. Just like a hapless worker pasting up sheets on a real billboard, your “virtual” sheets can be placed in the wrong location and may even be pasted upside down.



To complete this assignment you will need to use basic Python features and the Turtle graphics module. You must also design four billboard sheets which, when pasted in the correct locations and orientations, align to produce a single picture. The picture must be non-trivial, and must span all four sheets, but otherwise you have a free choice of what to draw, e.g.,

- an advertisement for a commercial product (real or fictional),
- a public service announcement,
- a political advertisement,
- corporate or sporting logos,
- a portrait of an individual (real or fictional),
- a landscape painting, or
- any other picture you like!

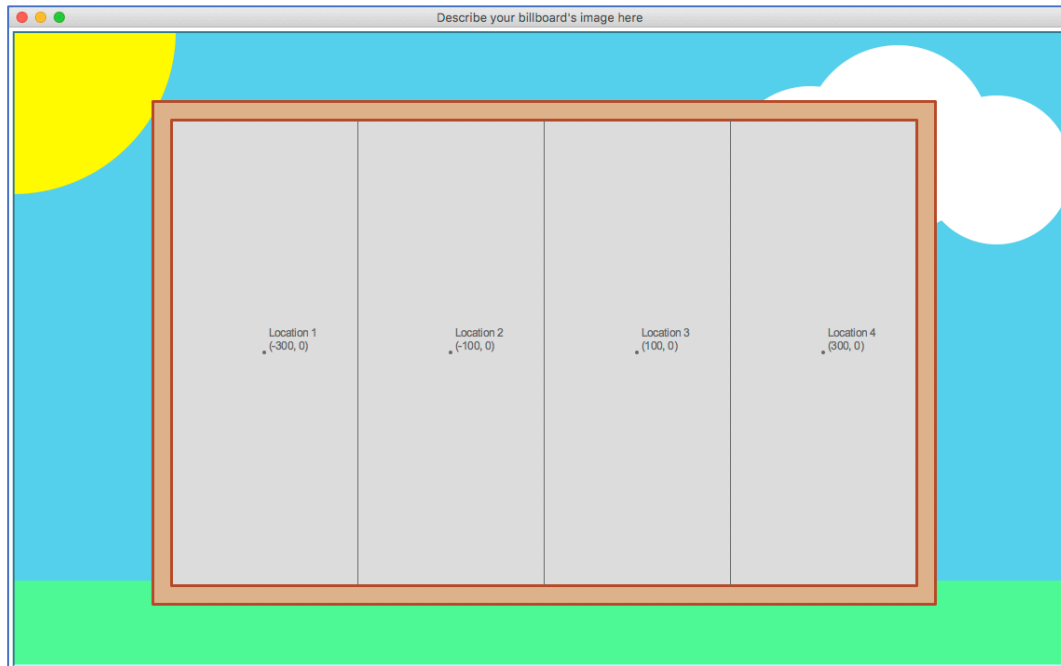


Most importantly, you are not limited to the types of images normally found on billboards. Any recognisable image will do!

To “paste” the individual sheets as specified in the given data sets, you must develop your code so that each sheet can be drawn in any of four different locations on the drawing canvas and in either of two different orientations. You should give thought to how to do this efficiently before you start drawing your image using Turtle graphics. Simply drawing an image at fixed coordinates on the canvas will not allow you to relocate and reorient it later on.

Resources provided

A template Python 3 program, `billboard.py`, is provided with these instructions. When run it creates a drawing canvas and displays a simple background image on which you will draw your billboard sheets. The image drawn by the initial template appears as follows.



The billboard's backing is divided into four locations for pasting up sheets. Each sheet must be a rectangle measuring 200×500 pixels. The centrepoint of each location has been marked to help you position your images as you develop your program. Your sheets must fit precisely into the four locations indicated.

The provided template file also contains several data sets, in the form of lists, which specify how you must arrange the sheets when drawing them. These instruction lists each contain two types of data:

- An initial value, either 'O' or 'X', whose purpose will be revealed only in the second part of the assignment. You should ignore it while developing your solution to Part A.
- Instructions saying how to paste up each individual sheet. Each such instruction is itself a list containing three values:
 - The name of the sheet to be pasted up, from 'Sheet A' to 'Sheet D'.
 - The location on the billboard's backing where the sheet is to be pasted, from 'Location 1' to 'Location 4'.
 - The orientation of the sheet, either 'Upright' or 'Upside down'.

There are thus many ways of pasting up the sheets, but to display the billboard image correctly Sheets A to D must be pasted upright in Locations 1 to 4, respectively. Any other arrangement will display the image incorrectly. Not all of the data sets mention all four sheets. If there is no instruction for a particular sheet then it should not be drawn at all.

The template file also contains a dummy function definition, `paste_up`. Your task is to complete this function definition so that when called with a data set as its argument it draws the sheets at the locations and in the orientations specified. The function must work for **any of the given data sets** or **any other similar data sets in the same format**.

Illustrative example

Here we present a sample solution to illustrate the requirement. (You should *not* copy our example! Develop your own idea! Be imaginative!)

Firstly you will need to design each of the four distinct rolled paper “sheets” that collectively form your image. In our case we have created the following four sheets, as shown below in their upright orientations.

Sheet A:



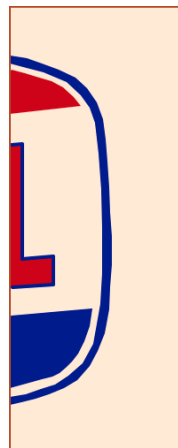
Sheet B:



Sheet C:



Sheet D:



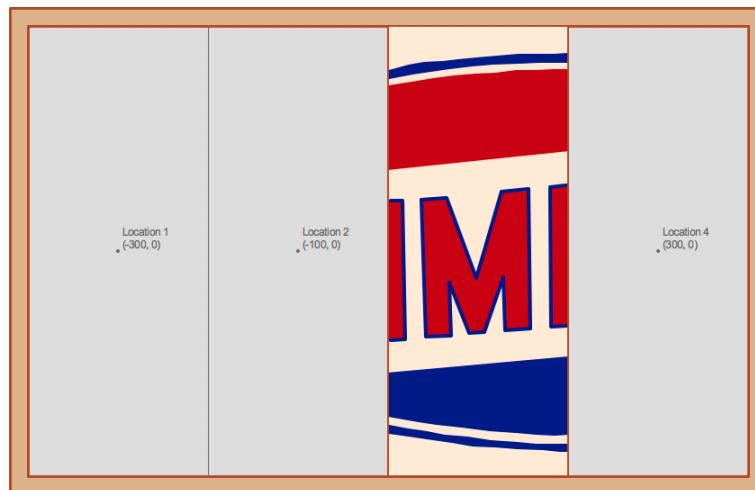
Each of the four sheets must contain a non-trivial image clearly distinct from all the others, regardless of their orientation, i.e., no sheet can be identical to another when turned upside down. Also, the images must be asymmetric so that it's easy to tell which way up the sheet has been pasted. When arranged correctly the four sheets must align precisely to produce a single composite picture.

Although it's difficult to specify the artistic requirements for this assignment, given the wide range of images that could be drawn, it's expected the assembled picture will involve several different shapes and colours and must be immediately recognisable. Simple geometric shapes would not be considered sufficiently challenging. Similarly, four unrelated images, one per sheet, would be unacceptable.

In the Python template file there are several data sets in the form of lists, each describing a particular way of pasting up one or more sheets. For instance, one of the simplest such lists is the one at index 12 in our data set, which is as follows:

```
['0', ['Sheet B', 'Location 3', 'Upright']]
```

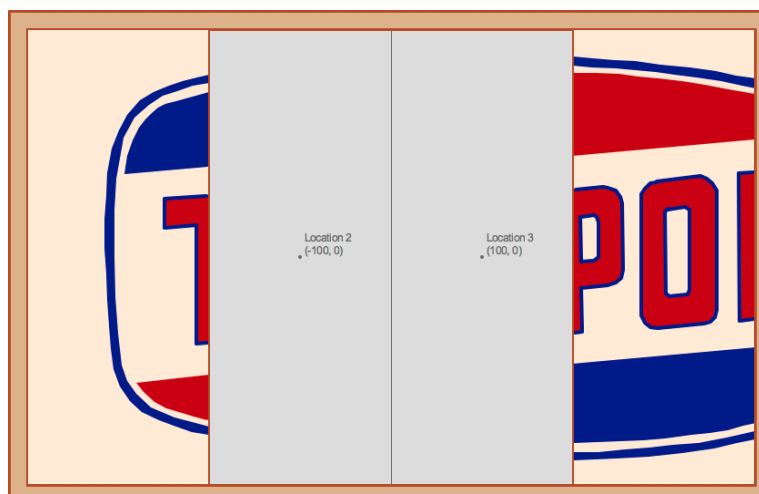
This list tells us that we are required to paste Sheet B in Location 3 in its upright orientation. When our `paste_up` function is called with this list as the argument it produces the billboard displayed below.



Another such data set is the one at index 36, as follows.

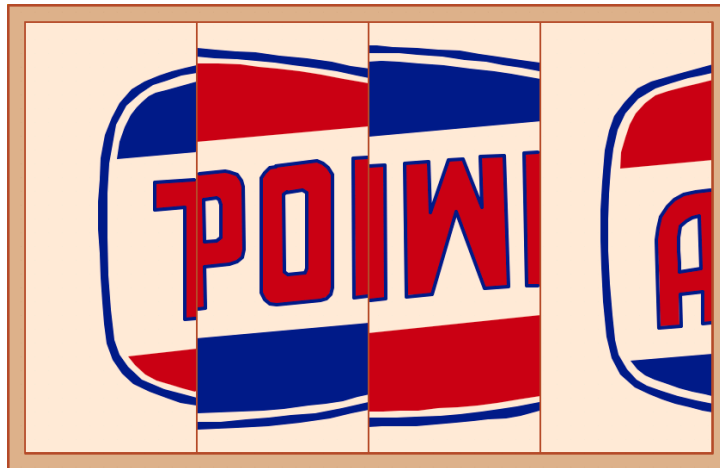
```
['0', ['Sheet D', 'Location 1', 'Upside down'],  
      ['Sheet C', 'Location 4', 'Upright']]
```

In this case Sheet D is to be pasted in Location 1, but upside down, and Sheet C is pasted in Location 4, in its correct orientation, which produces the following billboard.



Other data sets require all four sheets to be pasted up. For instance, the data set at index 47 specifies that four sheets must be displayed, but two are upside down as shown overleaf.

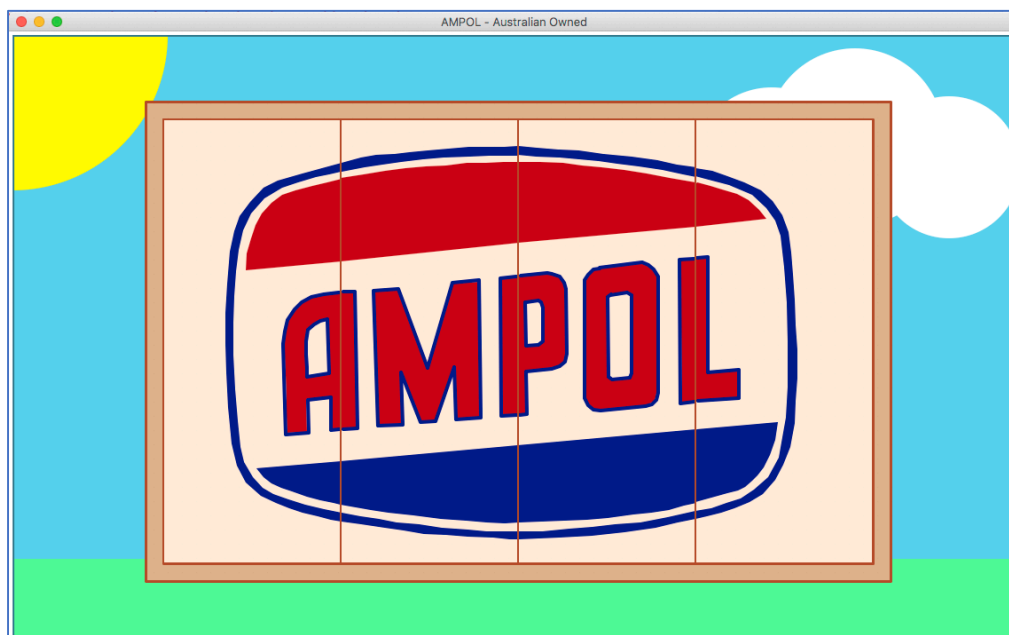
```
['0', ['Sheet C', 'Location 2', 'Upright'],
      ['Sheet B', 'Location 3', 'Upside down'],
      ['Sheet D', 'Location 1', 'Upside down'],
      ['Sheet A', 'Location 4', 'Upright']]
```



The final two data sets require all four sheets to be drawn, in their correct locations and orientations. The very last data set provided, at index 52, produces the correct billboard layout.

```
['0', ['Sheet A', 'Location 1', 'Upright'],
      ['Sheet B', 'Location 2', 'Upright'],
      ['Sheet C', 'Location 3', 'Upright'],
      ['Sheet D', 'Location 4', 'Upright']]
```

This data set pastes all four sheets upright and in locations that correctly complete the image, as shown below, finally revealing that our billboard displays the old AMPOL logo (see Appendix 2). We have also given the drawing canvas a title, describing the billboard.



Requirements and marking guide

To complete this part of the assignment you are required to extend the provided `billboard.py` Python file by completing function `paste_up` so that it can draw sheets at the locations and orientations specified by a data set provided as its single parameter. Your code must work for all the supplied data sets in the template file and **any other data set in the same format**.

Your submitted solution will consist of a single Python 3 file, and must satisfy the following criteria. Percentage marks available are as shown.

1. **Drawing four distinct billboard sheets (4%).** Your program must be able to draw four distinct sheets, each containing part of a single overall picture. Each sheet must be a 200×500 pixel rectangle and must be of a reasonable degree of complexity, involving multiple shapes and colours. The entire rectangle must be filled in with colour. Each sheet must be clearly distinct so that the viewer can tell which one has been drawn. It must be easy for the viewer to distinguish each sheet from its neighbours when they are side-by-side, either through the use of a visible margin or by contrasting background colours. Each sheet must be clearly different from all the others, regardless of their orientation, and the image on each sheet must be asymmetric so that it is easy to tell whether it is being displayed upright or upside down.
2. **Creating a single picture in four parts (3%).** When pasted up correctly, as per the final two data sets provided, the separate parts of the image must align perfectly to produce a single, clearly recognisable, non-trivial picture. The title of the drawing canvas must describe the picture drawn. (NB: If your solution is incapable of drawing all four sheets in their correct arrangement we cannot assess this criterion and must award zero for it.)
3. **Relocating sheets (4%).** Your code must be capable of drawing each of the four sheets at any of the four marked locations. The sheets must fit precisely into each of the locations marked on the billboard's backing, with the entire area being filled and no parts of the drawing protruding. The sheets must be sufficiently different that it is easy to distinguish which one is being drawn in each location. The images on the sheets must preserve their integrity no matter where they are drawn. Your solution for relocating the sheets must work for **all of the provided data sets** and **any other data sets in the same format**. You cannot "hardwire" your code for specific data sets and you may not change the data sets provided.
4. **Orienting sheets (5%).** Your code must be capable of drawing each of the four sheets in either of the two possible orientations, upright or upside down. The images on each sheet must be sufficiently asymmetric that it is easy to tell one orientation from the other. The images on the sheets must preserve their integrity no matter how they are oriented. Your solution for reorienting the sheets must work for **all of the provided data sets** and **any other data sets in the same format**. You cannot "hardwire" your code for specific data sets and you may not change the data sets provided.
5. **Code quality and presentation (4%).** Your program code, for both Parts A and B of the assignment, must be presented in a professional manner. See the coding guidelines in the *IFB104 Code Presentation Guide* (on Blackboard under *Assessment*) for suggestions on how to achieve this. In particular, given the obscure and repetitive nature of

the code needed to draw complex images using Turtle graphics, each significant code segment must be clearly commented to say what it does, e.g., “Draw letter ‘M’”, “Draw red oval”, etc.

6. **Extra feature (5%).** *Part B of this assignment will require you to make a ‘last-minute extension’ to your solution. The instructions for Part B will not be released until just before the final deadline for Assignment 1.*

You must complete the assignment using basic Turtle graphics and maths functions only. You may not import any additional modules or files into your program other than those already included in the given `billboard.py` template. In particular, you may not import any image files for use in creating your picture.

Finally, you are *not* required to copy the example shown in this document and you do *not* need to draw something that would be found on a “real” billboard. Instead you are strongly encouraged to be creative and to choose a picture that interests you personally. Surprise us!

Artistic merit – The Hall of Fame!

You will not be assessed on the artistic merit of your solution, only the ability to create a recognisable picture in four parts. However, a “Hall of Fame” containing the solutions considered the most artistic or ambitious by the assignment markers will be created on Blackboard. (Sadly, additional marks will not be awarded to the winners, only kudos.)

Development hints

- This can be a time-consuming task, so you are strongly encouraged to design your billboard sheets carefully before developing any program code.
- The hardest part of this assignment is the need to allow the sheets to be drawn in different locations and orientations. You therefore need to devise a way of drawing each sheet so that you can either (a) make all drawing moves *relative* to the starting position and orientation (using Turtle’s `forward`, `left` and `right` commands), or (b) calculate *absolute* target positions for each drawing move in terms of a given starting position and orientation (using Turtle’s `goto` command). Stick with one of these two approaches. Mixing these two styles of drawing will make the task harder.
- If you are unable to complete the whole assignment, just submit whatever parts you can get working. You will receive partial marks for incomplete solutions. Try to ensure that your program code runs when you submit it, even if it is incomplete.
- To help you debug your code we have provided several data sets, numbered 2 to 33, which draw just one sheet at a time. You can use these to help refine the code for each sheet separately.
- Part B of this assignment will require you to change your solution slightly in a short space of time. You are therefore encouraged to keep code maintainability in mind while developing your solution to Part A. Make sure your code is neat and well-commented so that you will find it easy to modify when the instructions for Part B are released.

Deliverable

You must develop your solution by completing and submitting the provided Python 3 file `billboard.py` as follows.

1. Complete the “statement” at the beginning of the Python file to confirm that this is your own individual work by inserting your name and student number in the places indicated. *We will assume that submissions without a completed statement are not your own work!*
2. Complete your solution by developing Python code to replace the dummy `paste_up` function. You must complete your solution using only the modules already imported by the provided template. You may *not* use or import any other modules to complete this program. In particular, you may *not* import any image files into your solution.
3. Submit *a single Python file* containing your solution for marking. Do *not* submit an archive (e.g., in ‘zip’ or ‘rar’ formats) containing several files. Only a single file will be accepted, so you cannot accompany your solution with other files or pre-defined images. **Do not submit any other files! Submit only a single Python 3 file!**

Apart from working correctly your program code must be well-presented and easy to understand, thanks to (sparse) commenting that explains the *purpose* of significant code segments and *helpful* choices of variable and function names. *Professional presentation* of your code will be taken into account when marking this assignment.

If you are unable to solve the whole problem, submit whatever parts you can get working. You will receive *partial marks for incomplete solutions*.

Plagiarism

This is an individual assessment item. All files submitted will be subjected to software plagiarism analysis using the MoSS system (<http://theory.stanford.edu/~aiken/moss/>). Serious violations of the university’s policies regarding plagiarism will be forwarded to the Science and Engineering Faculty’s Academic Misconduct Committee for trial.

How to submit your solution

A link will be available on Blackboard under *Assessment* for uploading your solution file before the deadline (5:00pm Wednesday, September 13th). You can *submit as many drafts of your solution as you like*. You are strongly encouraged to *submit draft solutions* before the deadline as insurance against computer or network problems near the deadline. Students who encounter problems uploading their Python files to Blackboard should contact the *IT Helpdesk* (ithelpdesk@qut.edu.au; 3138 4000) for assistance and advice.

Appendix 1: Some standard Turtle graphics colours you can use

Red colors				Green colors				Brown colors			
IndianRed	CD 5C 5C	205	92 92	GreenYellow	AD FF 2F	173	255 47	Cornsilk	FF F8 DC	255	248 220
LightCoral	FF 80 80	240	128 128	Chartreuse	7F FF 00	127	255 0	BlanchedAlmond	FF EB CD	255	235 205
Salmon	FA 80 72	250	128 114	LawnGreen	7C FC 00	124	252 0	Bisque	FF E4 C4	255	228 196
DarkSalmon	E9 96 7A	233	150 122	Lime	00 FF 00	0	255 0	NavajoWhite	FF DE AD	255	222 173
LightSalmon	FF A0 7A	255	160 122	LimeGreen	32 CD 32	50	205 90	Wheat	F5 DE B3	245	222 179
Crimson	DC 14 3C	220	20 60	PaleGreen	98 FB 98	152	251 152	BurlyWood	DE B8 87	222	184 135
Red	FF 00 00	255	0 0	LightGreen	90 EE 90	144	238 144	Tan	D2 B4 8C	210	180 140
FireBrick	B2 22 22	178	34 34	MediumSpringGreen	00 FA 9A	0	250 154	RosyBrown	BC 8F 8F	188	143 143
DarkRed	8B 00 00	139	0 0	SpringGreen	00 FF 7F	0	255 127	SandyBrown	F4 A4 60	244	164 96
Pink colors				MediumSeaGreen	3C B3 71	60	179 113	Goldenrod	DA A5 20	218	165 32
Pink	FF C0 CB	255	192 203	SeaGreen	2E 8B 57	46	139 87	DarkGoldenrod	B8 86 0B	184	134 11
LightPink	FF B6 C1	255	182 193	ForestGreen	22 8B 22	34	139 34	Peru	CD 85 3F	205	133 63
HotPink	FF 69 B4	255	105 180	Green	00 80 00	0	128 0	Chocolate	D2 69 1E	210	105 30
DeepPink	FF 14 93	255	20 147	DarkGreen	00 64 00	0	100 0	SaddleBrown	8B 45 13	139	69 19
MediumVioletRed	C7 15 85	199	21 133	YellowGreen	9A CD 32	154	205 50	Sienna	A0 52 2D	160	82 45
PaleVioletRed	DB 70 93	219	112 147	OliveDrab	6B 8E 23	107	142 35	Brown	A5 2A 2A	165	42 42
Orange colors				Olive	80 80 00	128	128 0	Maroon	80 00 00	128	0 0
LightSalmon	FF A0 7A	255	160 122	DarkOliveGreen	55 6B 2F	85	107 47	White colors			
Coral	FF 7F 50	255	127 80	MediumAquamarine	66 CD AA	102	205 170	White	FF FF FF	255	255 255
Tomato	FF 63 47	255	99 71	DarkSeaGreen	8F BC 8F	143	188 143	Snow	FF FA FA	255	250 250
OrangeRed	FF 45 00	255	69 0	LightSeaGreen	20 B2 AA	32	178 170	Honeydew	F0 FF F0	240	255 240
DarkOrange	FF 8C 00	255	140 0	DarkCyan	00 8B 8B	0	139 139	MintCream	F5 FF FA	245	255 250
Orange	FF A5 00	255	165 0	Teal	00 80 80	0	128 128	Azure	F0 FF FF	240	255 255
Yellow colors				Blue/Cyan colors				AliceBlue	F0 F8 FF	240	248 255
Gold	FF D7 00	255	215 0	Aqua	00 FF FF	0	255 255	GhostWhite	F8 F8 FF	248	248 255
Yellow	FF FF 00	255	255 0	Cyan	00 FF FF	0	255 255	WhiteSmoke	F5 F5 F5	245	245 245
LightYellow	FF FF E0	255	255 224	LightCyan	E0 FF FF	224	255 255	Seashell	FF F5 EE	255	245 238
LemonChiffon	FF FA CD	255	250 205	PaleTurquoise	AF EE EE	175	238 238	Beige	F5 F5 DC	245	245 220
LightGoldenrodYellow	FA FA D2	250	250 210	Aquamarine	7F FF D4	127	255 212	OldLace	FD F5 E6	253	245 230
PapayaWhip	FF EF D5	255	239 213	Turquoise	40 E0 D0	64	224 208	FloralWhite	FF FA F0	255	250 240
Moccasin	FF E4 B5	255	228 181	MediumTurquoise	48 D1 CC	72	209 204	Ivory	FF FF F0	255	255 240
PeachPuff	FF DA B9	255	218 185	DarkTurquoise	00 CE D1	0	206 209	AntiqueWhite	FA EB D7	250	235 215
PaleGoldenrod	EE E8 AA	238	232 170	CadetBlue	5F 9E A0	95	158 160	Linen	FA F0 E6	250	240 230
Khaki	F0 E6 8C	240	230 140	SteelBlue	46 82 B4	70	130 180	LavenderBlush	FF F0 F5	255	240 245
DarkKhaki	BD B7 6B	189	183 107	LightSteelBlue	B0 C4 DE	176	196 222	MistyRose	FF E4 E1	255	228 225
Purple colors				PowderBlue	B0 E0 E6	176	224 230	Gray colors			
Lavender	E6 E6 FA	230	230 250	LightBlue	AD D8 E6	173	216 230	Gainsboro	DC DC DC	220	220 220
Thistle	D8 BF D8	216	191 216	SkyBlue	87 CE EB	135	206 235	LightGrey	D3 D3 D3	211	211 211
Plum	DD A0 DD	221	160 221	LightSkyBlue	87 CE FA	135	206 250	Silver	C0 C0 C0	192	192 192
Violet	EE 82 EE	238	130 238	DeepSkyBlue	00 BF FF	0	191 255	DarkGray	A9 A9 A9	169	169 169
Orchid	DA 70 D6	218	112 214	DodgerBlue	1E 90 FF	30	144 255	Gray	80 80 80	128	128 128
Fuchsia	FF 00 FF	255	0 255	CornflowerBlue	64 95 ED	100	149 237	DimGray	69 69 69	105	105 105
Magenta	FF 00 FF	255	0 255	MediumSlateBlue	7B 68 EE	123	104 238	LightSlateGray	77 88 99	119	136 153
MediumOrchid	BA 55 D3	186	85 211	RoyalBlue	41 69 E1	65	105 225	SlateGray	70 80 90	112	128 144
BlueViolet	8A 2B E2	138	43 226	MediumBlue	00 00 CD	0	0 205	Black	00 00 00	0	0 0
DarkViolet	94 00 D3	148	0 211	DarkBlue	00 00 8B	0	0 139				
DarkOrchid	99 32 CC	153	50 204	Navy	00 00 80	0	0 128				
DarkMagenta	8B 00 8B	139	0 139	MidnightBlue	19 19 70	25	25 112				
Purple	80 00 80	128	0 128								
Indigo	4B 00 82	75	0 130								
SlateBlue	6A 5A CD	106	90 205								
DarkSlateBlue	48 3D 8B	72	61 139								
MediumSlateBlue	7B 68 EE	123	104 238								

Appendix 2: Irrelevant historical note

For our demonstration solution we draw the AMPOL logo as an example of something you might have seen on an advertising billboard while driving around Australia some years ago. But just what is AMPOL?

AMPOL, the Australian Motorists Petrol Company, was established in 1936 and at its peak operated service stations around the country. Its logo was widely seen everywhere, typically accompanied by the words “Australian Owned”, “Buy Australian”, or some other such patriotic message.



The company logo changed considerably over the years, but typically consisted of the name “AMPOL” with thick coloured stripes both above and below. The colours were usually red and blue on a white background. For our demo we’ve used the “classic” oval-shaped logo which was common in the late 1950s and early 1960s.



Sadly, like so many Australian companies, AMPOL was devoured by foreign interests in the early 1990s, in this case the American oil company Caltex. AMPOL signage on service stations is therefore rare these days, although you may run across surviving examples in isolated country towns.