Solution Notes - CGIP paper 3, 2003

(a) Lab

L - luminance, how light the colour is;
a,b - chrominance, one is red-green, the other is yellow-blue
Lab is an attempt to produce a perceptually-uniform colour space so it is best suited to uses where colours are being blended together or where we wish to do perceptual colour comparisons

CMYK

the proportions of the four inks used in colour printing
Cyan, Magenta, Yellow and Black (or Key)
CMYK is useful when printing as it matches the physical implementation of colour printers

HLS

H - hue, the dominant colour
L - luminance or lightness
S - saturation, how deep or saturated the colour is
HLS is a human-friendly colour space, useful in user interfaces

(b) a variety of methods exist
- the one taught in lectures uses one bit to distinguish between runs of identical and non-identical pixels, seven bits to give the length of the run (making a whole byte) followed by one byte giving the value of the identical pixels or the appropriate number of bytes giving the values of non-identical pixels.

(c)
```
function InBoundingRect (x1, y1, x2, y2, x3, y3, x4, y4 : float)
    returns integer
begin
    float minX = min (x1, x2, x3, x4);
    float minY = min (y1, y2, y3, y4);
    float maxX = max (x1, x2, x3, x4);
    float maxY = max (y1, y2, y3, y4);

    if (minX ≥ left & maxX ≤ right & minY ≥ bottom & maxY ≤ top)
    then return 0; /* completely inside */
    else if (minX > right | maxX < left | minY > top | maxY < bottom)
        then return 2; /* completely outside */
        else return 1; /* ambiguous */
end;

function DrawUnclippedBezier (float x1, y1, x2, y2, x3, y3, x4, y4)
begin
    if NearlyStraight (x1, y1, x2, y2, x3, y3, x4, y4)
    then DrawUnclippedLine (x1, y1, x4, y4);
    else
        begin
            DrawUnclippedBezier (~~~~);
            DrawUnclippedBezier (~~~~);
        end;
end;
```

these contain the same calculations as in the equivalent function calls in the question.

* NOTE: there is a nastiness to this question: the student must work out what each return value means; this is easy if they paid attention in lectures, but hard otherwise...

Marking Scheme

(a) 2 marks for explaining the dimensions $2 \times 2 = 4$
1 mark for explaining the uses $2 \times 1 = 2$
6

(b) any run-length encoding method is acceptable so difficult to separate out marks for specific features. A complete & correct answer will get full marks
6

(c) InBoundingRect
find minX, minY, maxX, maxY 1
if completely inside – correct test 1
– return 0 ½
if completely outside – correct test 1
– return 2 ½
ambiguous case – return 1 1
5

DrawUnclippedBezier
Nearly Straight 1
DrawLine 1
DrawUnclippedBezier ×2 1
3

8

20

CGIP (Computer Graphics & Image Processing)
Overall Scheme for Exam Questions 2003

The course has four sections:

Background	3 lectures
2D Computer Graphics	4 lectures
3D Computer Graphics	5 lectures
Image Processing	3 lectures

These must all be tested in exam, preferably in proportion to the number of lectures given

Question	Slides in 2000 Notes	Marks Allocated BG	2D	3D	IP
p3 (a)	37 (29-38)	6			
(b)	269-271 (258-290)				6
(c)	78-84, 110-111		8		
p5 (a)	173-178 (161-179)			10	
(b)	17-26, 39-40	4			
(c)	246-250				6
p6 (a)	101-109, 124-133		1	11	
(b)	98-100		8		
		10	17	21	12

These questions give a good spread across the whole syllabus, test both algorithms and more general knowledge & require a student to understand at least two disparate parts of the course in order to get good marks on any of the questions.