Computer Graphic, & Image Processing

(a) $\sum_i I_i . k_d . (\underline{L_i} . \underline{N})$ models diffuse (Lambertian) reflection. It is accurate for Lambertian reflectors & a reasonable approximation for other diffuse reflectors.

$\sum_i I_i . k_s . (\underline{R_i} . \underline{V})^n$ models specular reflection. It is an approximation with little physical basis, but it is good for plastics. It behaves less well for more complex specular reflectors, such as metals.

$I_a . k_a$ "models" all the remaining illumination — principally illumination caused by inter-reflection between surfaces. It is a gross approximation bearing no relationship to reality and must be 'tweaked' to make the image look right.

I = calculated illumination at point in question
I_a = "ambient illumination", a global fiddle factor
i = index of lights; it enumerates each light with a unique ID
I_i = the intensity of light i at the point in question
k_a = ambient reflection coefficient of the object in question
k_d = diffuse " " " " " " " " "
k_s = specular " " " " " " " " "
L_i = unit vector pointing at light i from the point in question
N = unit normal vector at " " " "
V = unit vector pointing to the camera from " " " "
R_i = unit vector pointing in the direction that light would be perfectly reflected, from light i, at the point in question
n = Phong's "specular coefficient" which determines the spread of the specular reflection

(b) · ray tracing renders by shooting a ray through each pixel in turn, seeing which object is hit first by the ray, and then calculating the illumination as in (a)

· z-buffer renders by drawing each polygon in turn, possibly with various illumination calculations, such as Gouraud or Phong shading

· both can be extended to anti-aliasing, ray tracing by shooting multiple rays through each pixel & averaging, z-buffer by extending to the A-buffer

· ray tracing allows for perfect reflection & refracting & shadows as part of the basic mechanism; each of these requires clever tricks & extensions to implement in z-buffer

· z-buffer draws only polygons; ray tracing can draw any object which can have a ray-object intersection routine written for it

CONTENT
This question tests the "3D computer graphics" part of the course

PRELIMINARY MARKING SCHEME

(a)
diffuse – what it models	1	D ✓
what accuracy	1	✓
specular – what it models	1	S ✓
what accuracy	1	✓
ambient – what it models	1	A ✓
what accuracy	1 6	✓

unit vectors	1	U ✓
Li N	1	L V
Ri V	1	R ✓
n	1	N
ka kd ks	1	K
I Ia Ii ·	1 6 12	I

(b) one mark for each of up to eight salient facts 8

20