Program Sketch for Ray Tracing

program raytrace
var lsou; (* intensity of light source *)
back; (* background intensity *)
ambi; (* ambient light intensity *)
depth; (* depth of ray tree consisting of multiple
reflection/refraction paths *)

ray = record                (* ray      x = a + ti
    point: (a, b, c)                y = b + tj
    direction: (i, j, k)            z = c + tk  *)
end;
r: ray;

function intensity (r);        (* intensity = spec + refr +dull
    spec = specular reflection component
    refr = refraction component
    dull = non-reflecting, non refracting
    component *)
    L: unit vector pointing to light source
    N: unit surface normal
    V:  unit viewing vector
    Objects [1...n] (* list of n objects in scene *)
    Ka [1...n]  (* ambient reflectivity factor for each object *)
    Ks [1...n]  (* specular reflectivity factor for each object *)
    Kr [1...n]      (* refractivity index for each object *)
    Kd [1...n]  (* diffuse reflectivity factor for each object *)
    ns [1...n]  (* shininess factor for each object *)
    (* Additional Comments: Ka[j] can be anything. For a transparent object,
    Kd[j]=0 and
    Ks[j]+kr[j]=1    i.e. partly reflecting + partly refracting
    For an opaque object Kr[j]=0,  Ks[j] and Kd[j] can be anything
    as no simple relation between them *)

function intensity(r: ray): rgb
var flec, frac: ray;    spec, refin, dull: rgb;
begin
    depth := depth +1
    if depth >5 then intensity :=back
    else
        begin (* label 1 *)
            check ray r for intersection with all objects in scene
            if no intersection
                then if r parallel to L
                    then intensity :=lsou
                    else intensity :=back
                else
                    begin (* label2 *)
                        Take closest intersection which is object[j]
                        compute normal N at the intersection point
                    end;
        end (* label 1 *)
end;
if $K_{s}[j] > 0$ (* non-zero specular reflectivity *)
then begin
    compute reflection ray flec;
    spec := $K_{s}[j] \times \text{intensity(flec)} \times (r \cdot V)^{n_{s}[j]}$;
end
else spec := 0;
if ($K_{r}[j] > 0$) (* non-zero refractivity *)
then begin
    compute refraction ray frac;
    refr := $K_{r}[j] \times \text{intensity(frac)}$;
end
else refr := 0;
check for shadow;
if shadow
    then dull := $K_{a}[j] \times \text{ambi}$
else dull := $K_{d}[j] \times \text{lso} \times N \cdot L + K_{a}[j] \times \text{ambi}$;
intensity := spec + refr + dull;
end (* label2 *)
end (* label 1 *)
depth := depth - 1
end (* function *).

begin (* raytrace *)
    for each pixel P of projection viewport in raster order
    begin
        r = unit ray emanating from viewer through P; V = r;
        set intensity(r) to the frame buffer pixel corresponding to P
    end
end (*raytrace *)