Student number	point total	req total	extra total	R0 UV (0.5p)	R1 ortho, ambient (1.5p)	R2 Depth vis (1p)	R3 Perspective (1.5p)	R4 Phong, lights (3p)	R5 Planes (1p)	R6 Triangles (1.5p)	R7 Shadows (1.5p)	R8 Reflection (1.5p)	R9 AA (2p)	mod	notes / extras / ...	Refractio n (1-2p)	Simple fog (1p)	More primitives (3p)	Arbitrary filters (1-3p)	Stereo cubemap (4p+)	transparent shadows (1.5p)	Fresnel (1p)	Texture s (2-4p)	Normal mapping (2-3p)	CSG (4-5p)	Other extras (?p)	What other extras
218096	0	0	0																								
225157	0	0	0																								
270034	4	4	0	0.5	1	1	1.5								R1: The scale of the images is now incorrect, you should divide the size by 2 in generateRay.												
292009	6.5	6.5	0	0.5	1.5	1	1.5		0.5	1.5					R5: Planes upside down, remove the multiplication by -1 when calculating t.												
292326	15	14	1	0.5	1	1	1.5	2.5	1	1.5	1.5	1.5	2		R4: specular added even if light below local horizon.											1	Randomized start for regular and jittered
292986	0	0	0																								
293545	1.5	1.5	0	0.5	1										R1: Ambient incorrect, you should multiply the diffuse color by the ambient intensity.												
295323	0	0	0																								
297606	0	0	0																								
311210	0	0	0																								
347022	0	0	0																								
350006	10.5	10.5	0	0	1	1	1.5	2	1	1.5	1.5	1			R0: Your inputs to FW::rcp are integers, therefore the function uses integer division for the rcp operation and you always get 0 as a result; R1: Size should be divided by 2 when generating rays; R4: You should return 0 when the light is below the local horizon. Point light dir_to_light incorrect, you should also inverse it for directional lights (which you do later in the shade function); R8: Epsilon used incorrectly. Input it to traceRay in place of tmin instead.												
350475	0	0	0																								
353692	6	6	0	0.5	1	1	1.5	1					1		R1: Y-axis inverted. You've "fixed" this for the perspective camera by inverting it there. Ambient light should be the diffuse color multiplied by ambient intensity; R9: Regular&Jittered: Replace the n-x with just n. Also divide the x&y by m_dim in jittered; R4: Directional light direction inverted. Point light distance to the light is just the length of the vector, no need for sqrt. Diffuse shading ok, shade back used incorrectly but no points taken here												
353757	0	0	0																								
357083	0	0	0																								
362256	0	0	0																								
401311	0	0	0																								
424615	17.5	14.5	3	0.5	1.5	1	1.5	3	1	1.5	1.5	1	2		R8 & Refraction: When calling traceRay, use a new Hit() as argument to get correct depth and normal images; Refraction: traceRay should always be multiplied with color, not only when entering object, and total internal reflection is readily handled by the reflection code path (so in reflection there shouldn't be a condition and during total internal reflection you shoudn't do anything). More primitives: Transform only	1.5		1.5									
425494	0	0	0																								
425614	8.5	8.5	0	0.5	1.5	1	1.5	3	1						R4: Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal;												
426419	0	0	0																								
426736	0	0	0																								
427492	0	0	0																								
427793	15	15	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Good points about the handout.												
427845	0	0	0																								
428381	0	0	0																								
428789	3	3	0	0.5	1	1	0.5								R1: Objects too small in images. Offset from center should be divided by two; R3: Several mistakes in direction;												
430324	0	0	0																								
430463	1	1	0	0.5	0.5										R1: Missing ambient light. Y-axis inverted. Not taking into account camera orientation, only works when aligned with z-axis.												
431857	11.5	11.5	0	0.5	1.5	1	1.5	3	1	1.5	1.5																
432241	8.5	8.5	0	0.5	1.5	1	1.5	3	1																		
437631	1	1	0	0.5	0.5										R1: point.x should scale the horizontal and point.y the upward direction. The direction of the camera can be anywhere in 3D, not just in 2D, and the image plane is now infinitly far away from the origin where most of the objects are. y-axis should be flipped. No ambient lighting.												
438397	0	0	0																								
472379	0	0	0																								
473158	0	0	0																								
473420	0	0	0																								
473637	0	0	0																								
474380	0	0	0																								
475389	0	0	0																								
475758	0	0	0																								
475910	0	0	0																								
476498	0	0	0																								
477170	0	0	0																								
477400	13.5	13.5	0	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	1.5		R4: Specular should be 0 when the light is below the local horizon. The point light intensity should be divided by the attenuation terms, not multiplied; R9: Regular sampler incorrect												
477617	2	2	0	0.5	0.5	1									R1: Should flip y-axis, not x-axis. Ray generation incorrect. Ambient light should be scene.ambient * material.diffuse; R2: Correct. R1 errors cause the black screen here.												
477659	0	0	0																								
477701	7.5	7.5	0	0.5	1.5	1	1.5	3																			
478328	0	0	0																								
478470	0	0	0																								
478687	0	0	0																								
479505	0	0	0																								
479576	0	0	0																								
479725	1	1	0	0.5	0.5										Readme not filled!												
480248	0	0	0																								
480303	13.5	12.5	1	0.5	1.5	1	1.5	3	1	1.5	1.5	1			R8: Otherwise correct but the result isn't added to the total lighting; R9: Claimed but no implementation found; Transform: Stack overflow, call object_->intersect instead of the transform intersect recursively. Don't normalize the homogeneous vector, transform back to cartesian first.			1									
480730	81	14	67	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	1.5		R0: crashes with empty scene (due to div by 0 for the perf print), R4: specular added even if light below local horizon, R9: all pixels use the same random patterns. Nice work with the SSE tracer!	2		3	3			1	4	3		51	SSE tracer with SAH BVH (15p), instancing (3p), DOF (3p), area lights (5p), path tracing with roulette and material importance sampling (10p), BVH visualization (4p), tiled rendering (2p), aniso and FELINE (4p), other brdf (modified phong, 1p), HDR skybox (2p), tone mapping (2p)
481014	15	15	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2														
481441	3	3	0	0.5	1	1	0.5								Both camera models are relatively far off; there is no need to handle different ranges differently.												
493578	0	0	0																								
506355	6.5	6.5	0	0.5	1	1	1.5	2.5							R1: ortho camera origin and direction computed but not returned. R4: specular added even if light below local horizon												
508793	0	0	0																								
514020	0	0	0																								
516109	1.5	1.5	0	0.5	1										R1: The scale of the images is now incorrect, you should divide the size by 2 in generateRay. Don't subtract 1 from the imageSize, the images are now very slightly offset;												
519656	0	0	0																								
525653	0	0	0																								
525666	13	13	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1	1		R4: shade_back is a cmd line argument and it is not used in shade. Distace of point light is always 1; R8: Parameter tmin for traceRay should be 0, not tmin. Reflections with orthographic camera look incorrect because of this; R9: In regular sampler you have the division by two in the wrong place, it should be (x + 1.0/2) and (y + 1.0/2). Jittered sampler incorrect. It is similar to regular except instead of a 0.5, you add a random number between 0 and 1.												
525792	0	0	0																								
525925	14.5	13.5	1	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	0.5		R4: shade_back is a cmd line argument; R9: You weren't supposed to add a loop in main.cpp, the inner loop already goes through num_samples. No anti-aliasing occurs because the same ray is used in the new inner loop when assigning values to sample_color. Regular sampler does not go through nth subpixel. Jittered sampler also incorrect; Refraction: Not taking into account whether ray is going into or out of the object. Should use transparent color, not reflective color.	1											
526490	16.5	15	1.5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Primitives: Transform only			1.5									
526717	12.5	12.5	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5			R4: Specular should be 0 when the light is below the local horizon;												
526746	0	0	0																								
527143	0	0	0																								
527347	3	3	0	0.5	1.5		1								R3: Perspective camera incorrect. Origin should just be the center, direction comes from direction, up and horizontal vectors along with the pixel position and fov angle.												
527389	0	0	0																								

Student number	point total	req total	extra total	R0 UV (0.5p)	R1 ortho, ambient (1.5p)	R2 Depth vis (1p)	R3 Perspective (1.5p)	R4 Phong, lights (3p)	R5 Planes (1p)	R6 Triangles (1.5p)	R7 Shadows (1.5p)	R8 Reflection (1.5p)	R9 AA (2p)	mod	notes / extras / ...	Refraction (1-2p)	Simple fog (1p)	More primitives (3p)	Arbitrary filters (1-3p)	Stereo cubemap (4p+)	transparent shadows (1.5p)	Fresnel (1p)	Textures (2-4p)	Normal mapping (2-3p)	CSG (4-5p)	Other extras (?p)	What other extras
527444	0	0	0																								
527923	20	15	5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2			2		3									
528634	10.5	10.5	0	0.5	1.5	1	1.5	2	1	1.5	1.5		0		R4: If shade_back is true, you should inspect the the direction of the incoming ray with relation to the normal, not calculate the length of their difference. Specular added even if light is below local horizon. You have defined dir_to_light in directional light as direction even though it should be -direction. As a result, you have negated dir_to_light in shade. Because the dir_to_light in point lights point in the correct direction, point lights do not work. In point lights, you have a typo when calculating the attenuation.										
528883	0	0	0																								
529293	0	0	0																								
529303	0	0	0																								
529617	0	0	0																								
529992	20	15	5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		No test scene and box parsing provided! Box intersect doesn't seem to work.	2	1	2									
530185	0	0	0																								
530907	7	7	0	0.5	1.5	1	1.5	0	1	1.5					R4: directional light has no position, so computing the difference makes little sense. R6: the issue with the triangle normals is not normalizing them; this would be more apparent when working on the shading.												
530981	5.5	5.5	0	0.5	1	1	1.5	1.5							R1: camera scaling off by factor of 2, R4: using length(a*b) instead of dot(a,b) for dot products, incoming light direction is incorrect (it's the opposite to the direction of the light)												
540094	0	0	0																								
540311	13	13	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5			R1: You would get the exact same images as the example if you removed -1 from imageSize.x and imageSize.y., R4: Shade_back not used												
541543	0	0	0																								
544375	21	15	6	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Transform: Normal not normalized; Filtering: When adding samples you should consider which pixel centers are in the support radius. Not just whether the support reaches some part of a pixel. Thread safety can be achieved with a simple mutex around the line where you do read&write. Gaussian filter missing sigma in evaluation.	2		2.5	1.5								
544566	0	0	0																								
549749	0	0	0																								
552969	0	0	0																								
556347	0	0	0																								
561578	0	0	0																								
563068	0	0	0																								
570116	0	0	0																								
586210	20	14.5	5.5	0.5	1.5	1	1.5	3	1	1.5	1.5	1	2		R8 & Refraction: Parameter tmin for traceRay should be 0, not tmin. Reflections/Refractions with orthographic camera look incorrect because of this; Refraction: Total internal reflection is readily handled by the reflection code path (so if transmittedDirection returns false you shouldn't do anything!); Fog: Would look a lot better if fog color is returned when ray does not hit anything.	2	0.5	3									
587170	18.5	14.5	4	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1.5		R4: Normal should be flipped when shade_back is true; R9: Remove 0.5f from jittered sampler; Refraction: No need for the reflect else; Box: No parser implemented so boxes cannot be drawn. Intersect does not seem to work;	2		2									
587921	20	15	5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Transform: Normal not normalized; Refraction: You should only trace one reflection and one refracted ray per hit. Now you always trace one reflection ray and after that another along with the refracted ray.	1.5		2.5				1					
588137	0	0	0																								
588441	3.5	3.5	0	0.5	1	0.5	1.5								R1: Y-axis inverted; R2: Missing clamp and inversion of the range to [1,0]; R3: The image is upside down because of the mistake in R1 and you attempting to fix it here by flipping the x-axis. It happens to look correct in the test because the test case is symmetric!												
589291	0	0	0																								
589437	15.5	13.5	2	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	1.5		R4: specular added even if light below local horizon, shade_back not implemented. Point light quadratic attenuation should have the coefficient outside the power (we want a*(r^2), not (a*r)^2!) R9: jittered generates numbers from 0 to 9 but divides by 100 so the range is [0;0.1] instead of the desired [0;1].			2									
589848	0	0	0																								
590112	6.5	6.5	0	0.5	1	1	1.5	1	0.5	1					R1: the coordinate returned from normalizedImageCoordinateFromPixelCoordinate should be negated. R4: diffuse term is correct in principle but not used in the renderer itself. R5: plane normal is just normal_; the offset shouldn't have a minus sign in the formula. R6: triangle normal is normalized cross product of v01, v02.												
590332	11.5	11.5	0	0.5	1.5	1	1.5	1.5	1	1.5	1.5	1	0.5		R4: specular added even if light below local horizon (if diffuse component is 0, light is not visible; should have 0 specular as well), directional light direction (and thus the shading code) inverted; dir_to_light should be opposite to light direction -- would be fine but point lights use the correct direction R8: should offset point or use non-zero tmin in order to avoid numerical issues R9: no randomness in either uniform or jittered sampling and the rendering loop doesn't accumulate the samples												
590426	15	15	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2														
593177	0	0	0																								
593452	11.5	11.5	0	0.5	1.5	0.5	1.5	2.5	1	1.5	1	1.5			R2: You should subtract args.min from hit.t in the formula. Also missing clamping; R4: Return 0 when dot(N,L) <= 0. Specular should use dot (dir_to_light, reflection); R7: Should use continue statement instead of break in the light loop.												
593876	10	10	0	0.5	1.5	1	1.5	3	1	1.5					The objects appear in the plane scenes only after implementing the transformation extra.												
594367	0	0	0																								
594590	0	0	0																								
594930	0	0	0																								
595201	19	15	4	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2			2		2									
595612	13	13	0	0.5	1.5	1	1	3	1	1.5	1	1.5	1		R3: 0.5f should be only in fov_angle*0.5f. This causes the white in the bunny images; R7: Shadows shouldn't be enabled unless specified so by args_.shadows; R9: In uniform sampler you are sampling from a normal distribution, in jittered sampler there shouldn't be a 2.0f and in regular sampler the samples do not go through the Nth subpixel.												
596048	4	4	0	0.5	1	1	1.5								R1: point[0] and point[1] switched in generateRay()												
596242	14.5	14.5	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1	2		R0: Because of the way you have implemented R9, sample_color shouldn't have += but =, and it should be multiplied by num_samples to get the correct result; R4: Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal; R8: Parameter tmin for traceRay should be 0, not tmin. Reflections with orthographic camera look incorrect because of this;												
596747	0	0	0												R1: Don't subtract 1 from the imageSize, the images are now very slightly offset;												
596789	0	0	0																								
596792	16.5	14.5	2	0.5	1	1	1.5	3	1	1.5	1.5	1.5	2			2											
596857	2	2	0	0.5	1	0.5									R1: You shouldn't add 0.5f to the pixel coordinate, it's already done before the function is called! You should add the camera center to the ray origin, now your ortho camera is always located in the origin; R2: Incorrect order of calculations with the parentheses. Should be 1-(t-min)/(max-min)												
597445	17	15	2	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2			2											
598088	16.5	14.5	2	0.5	1.5	1	1.5	3	1	1.5	1	1.5	2		R7: The tmin for shadow rays should be directly set to a small value (tmin will be -FLT_MAX for orthographic cameras!)	2											
598318	0	0	0																								
602851	0	0	0																								
603067	9	9	0	0.5	1.5	1	1.5	1.5	1	1	0.5	0.5			R4: Directional light direction should be inverted. Shade_back used incorrectly (no points taken). Diffuse shading ok but you should care for negative dot(N,L); R6: t is incorrect. You did it correctly for the plane earlier; R7: Call shade only once, i.e. when the object isn't shadowed. Nudge the starting point of the shadow ray along the normal by some small epsilon, not tmin. Here the light direction shouldn't be inverted; R8: Mirror the old ray direction with the normal to get reflected ray direction. Reflected ray origin should just be the previous hit point. Again, you need some epsilon to avoid acne, you can give it as a parameter to traceRay instead of tmin. The answer should be incremented by the return value of traceRay multiplied by the reflective color of the surface.												
603096	0	0	0																								
603326	0	0	0																								

Student number	point total	req total	extra total	R0 UV (0.5p)	R1 ortho, ambient (1.5p)	R2 Depth vis (1p)	R3 Perspective (1.5p)	R4 Phong, lights (3p)	R5 Planes (1p)	R6 Triangles (1.5p)	R7 Shadows (1.5p)	R8 Reflection (1.5p)	R9 AA (2p)	mod	notes / extras / ...	Refraction (1-2p)	Simple fog (1p)	More primitives (3p)	Arbitrary filters (1-3p)	Stereo cubemap (4p+)	transparent shadows (1.5p)	Fresnel (1p)	Textures (2-4p)	Normal mapping (2-3p)	CSG (4-5p)	Other extras (?p)	What other extras
604105	16	13.5	2.5	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	1.5		R0: Does not draw anything after changes made for R9; R4: shade_back is a cmd line argument. Distance not defined for directional light. Point light has dir_to_light flipped and brackets missing from the denominator of the attenuation. The idea was to calculate the diffuse lighting also in shade(). Shade_back is about the direction of the incoming ray with relation to the normal, it has nothing to do with vr. If ray hits the surface from behind and shade_back is true then normal should be flipped always, not only when calculating r; R9: The offset in jittered sampler is too small, should be multiplied by two; Refraction: Total internal reflection is readily handled by the reflection code path; Fog: Would look a lot better if fog color is returned when ray hits nothing;	2	0.5										
606064	0	0	0																								
606268	0	0	0																								
608952	6.5	6.5	0	0.5	1.5	1	1.5	1.5	0.5						R4: In point lights, distance is always 1, dir_to_light is not normalized and you should multiply attentuation with intensity_ when calculating incident_intensity. Shade_back is a cmd line argument. clampedSpecular incorrect. Specular light added even if light is below local horizon; R5; All the conditions regarding t need to be true for you to assign values for h and return true. Also, always use this->material() and this->normal_ to get the values belonging to the object, do not take them from h;												
609142	21	14	7	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1		R9: Regular and jittered samplers incorrect; Transform: Normal not normalized; Area light: Should have 1/r^2 attenuation. Sampling isn't uniform over the disc, it's more dense in the middle. MC integration requires that you divide f (x) by the probability density p(x). For uniform distribution this is the 1/area of the sampled shape. Transforming from disc to hemisphere doesn't seem quite correct. The probability density is also different over the hemisphere after transforming. Join the advanced class to learn more!		1	1								5	Area light(5p)
609155	0	0	0																								
609168	15	15	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Transform: Normal not normalized												
610827	4	4	0	0.5	1.5	1	1								R3: r not normalized												
612155	0	0	0																								
612540	0	0	0																								
612812	0	0	0																								
621308	0	0	0																								
647175	0	0	0																								
647502	20.5	14.5	6	0.5	1.5	1	1.5	3	1	1.5	1.5	1	2		R8 & Refraction: Parameter tmin for traceRay should be 0, not tmin. Reflections with orthographic camera look incorrect because of this; R9: The images move slightly because in the regular sampler 1 / 2 is always 0, use 1.0/2.0 instead.	2		3				1						
648080	14.5	14.5	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	2		R4: material reflects even when light below local horizon												
648569	0.5	0.5	0	0.5																							
648860	0	0	0																								
649458	0	0	0																								
650191	0	0	0																								
650560	0	0	0																								
650829	1.5	1.5	0	0.5	1										R1: y-coord flipped, orthographic camera points in constant direction												
651640	14.5	12.5	2	0.5	1.5	1	1.5	3	1	1.5	1.5	1			R4: Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal; R8 & Refraction: Parameter tmin for traceRay should be 0, not tmin. Reflections/Refractions with orthographic camera look incorrect because of this; Refraction: When going out of the material, normal should be flipped. Angle should be calculated with -N, not T, and criticalAngle should have 1/Nr, not Nr. Easier would have been to just check if sqrtInside >= 0; More primitives: Transform only. Normals are directions so they should have w = 0 and they should be normalized.	1		1									
651802	0	0	0																								
652209	20	14	6	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1		Shadows etc. are not visible when running render_all.bat because of your modifications to render_options.bat; R4: Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal; R9: Regular and jittered sampler incorrect; Box: No parsing implemented, so cannot draw boxes. In intersection, you should swap t01 and t02 if (t01 > t02), and remove the lines with min and max at the end of the loop; Filters: Not thread safe. Using != NAN does not work, becuase comparisons with NAN are always false;	2		2	2								
652584	18.5	14	4.5	0.5	1	1	1.5	2.5	1	1.5	1.5	1.5	2		R1: Y-axis slightly offset. R4: Specular should be 0 when the light is below the local horizon; Box: Normal transformed as a position.	2		2.5									
653156	0	0	0																								
653347	12	12	0	0.5	1	1	1.5	3	1	1	1.5	1.5			R1: You need to add camera center to the ortho ray origin; R6: Flip the normal												
654142	13.5	13.5	0	0.5	1	1	1.5	2.5	1	1.5	1.5	1.5	1.5		R4: Point light direction is opposite (from light to point) R9: your UniformSampler is what was supposed to be RegularSampler and your RegularSampler doesn't make much sense (it overblurs based on the filter size); uniformsampler should be uniform over the pixel (like jittered but with only the random component)												
654294	13	13	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5															
654618	14	14	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1	2		R4: shade_back not used, both d and s should be clamped to zero when d < 0; R8: Parameter tmin for traceRay should be EPSILON, not tmin+EPSILON. Reflections with orthographic camera look incorrect because of this. R9: The i is there as a parameter because the base class Sampler has it as parameter.												
655109	0	0	0																								
655361	0	0	0																								
655390	0	0	0																								
656014	7	7	0	0.5	1	0.5	0.5	3	0.5	1					R1: The normalized image coordinates should range from [-1,1]^2. Size should be divided by 2 in ortho camera. R2: The depth range should be inverted to [1,0]; R3: The distance to the image plane is given by d = 1/tan(fov/2), multiply direction by this and add the up and horizontal vectors; R5: You should subtract dot(N,Ro) from the offset, not add; R6: Offset should be calculated by dot product. d1,d2,d3 should be floats, not ints.												
657068	0	0	0																								
657181	14	14	0	0.5	1.5	0.5	1.5	2.5	1	1.5	1.5	1.5	2		R2: depth total uninitialized, giving undefined behavior (mostly just black here), R4: specular added even if light below local horizon, R7: distance to light not used (but not visible in any of the given scenes)												
657437	13.5	13.5	0	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	1.5		R4: specular added even if light below local horizon, no shade_back, point light distance wrong (should have sqrt) and attenuation should be divided instead of multiplied. R9: the x and y steps of regular sampler should be divided by m_dim, just as in the jittered sampler.												
657767	32.5	14.5	18	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1.5		R9: samplers hard coded for 9 samples. CSG exhibits some artifacts from extreme angles (for example, look at the intersected spheres from the side of the substracted sphere)	2	1	3				1			3	8	Area light(5p), Motion blur (3p)
657893	0	0	0																								
663434	0	0	0																								
665173	14	14	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1		R9: Regular sampler samples are offset by half a subpixel in both axes. Jittered sampler incorrect.												
665678	28	15	13	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		Torus has some issues, but the preview object is a nice touch.	2		3	3		1.5	1				2.5	Torus
666208	0	0	0																								
666211	15.5	13.5	2	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	1.5		R4: Specular incorrect. Shade_back does not work correctly because you return zero immediately if dot(dir_to_light, normal) >= 0 even though when shade_back is true you should flip the normal first; R9: Jittered sampler goes over the pixel. Remove +1 from i and j.	2											
666253	0	0	0																								
710015	14.5	14.5	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1.5		R9: Regular sampler y slightly incorrect (should be divided by m_dim!) and jittered missing.												
715298	19	14.5	4.5	0.5	1	1	1.5	3	1	1.5	1.5	1.5	2		R1: Size should be divided by 2 in the ray generation; Transform: Normal transformed as a position	2		2.5									
716734	0	0	0																								
717377	0	0	0																								
717539	0	0	0																								
718020	0	0	0																								
718208	14	14	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1	2		R4: shade_back is a cmd line argument. Condition before if(shade_back) should have dot product > 0. Specular added even if light is below local horizon. Distance not defined for directional light. R8: Parameter tmin for traceRay should be 0, not tmin. Reflections with orthographic camera look incorrect because of this.												
718512	12.5	12.5	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1			R4: shade_back not used in shade(); R8: You should use a new Hit() in traceRay to get the correct depth and normal images.												

Student number	point total	req total	extra total	R0 UV (0.5p)	R1 ortho, ambient (1.5p)	R2 Depth vis (1p)	R3 Perspective (1.5p)	R4 Phong, lights (3p)	R5 Planes (1p)	R6 Triangles (1.5p)	R7 Shadows (1.5p)	R8 Reflection (1.5p)	R9 AA (2p)	mod	notes / extras / ...	Refraction (1-2p)	Simple fog (1p)	More primitives (3p)	Arbitrary filters (1-3p)	Stereo cubemap (4p+)	transparent shadows (1.5p)	Fresnel (1p)	Textures (2-4p)	Normal mapping (2-3p)	CSG (4-5p)	Other extras (?p)	What other extras
718826	25.5	14.5	11	0.5	1	1	1.5	3	1	1.5	1.5	1.5	2		R1: Don't subtract 1 from the imageSize, the images are now very slightly offset; Fog would look better if you just returned the fog color when you miss the scene.	2	0.5	3	3		1.5	1					
719032	14	14	0	0.5	1.5	1	1.5	2.5	1	1.5	1	1.5	2		R4: Quadratic attenuation term incorrect, should be qa*d^2, not (qa*d)^2; R7: Shadows shouldn't be on unless specified so by args_.shadows												
721619	0	0	0																								
721923	0	0	0																								
723154	0	0	0																								
723329	13	13	0	0.5	1.5	1	1.5	2	1	1.5	0.5	1.5	2		R4: directional light has inverse direction (direction towards the light is opposite to the illumination direction!) and consequently the material assumes an inverted direction. Speculars added even if light below horizon. R7: should cast a ray to determine if light is visible or not.												
723468	14.5	14.5	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	2		R4: Point light direction is opposite (from light to point)												
723484	12	12	0	0.5	1.5	1	1.5	2.5	1	1.5	1	1.5	0		R0: should divide x coord by args.width, not args. weight R4: Point light direction is opposite (from light to point) R8: shadow hit extent should be 'distance' (see, for example, the 'exponent variation' scenes with shadows on), R9: AA claimed but nothing implemented												
723565	15	15	0	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2														
723976	0	0	0																								
724483	0	0	0																								
726915	14.5	14.5	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	2		R4: specular added even if light below local horizon, R9: regular sampler generates same position for all samples (part = 1 / m_dim does an int division; should cast either operand to float)												
728696	0	0	0																								
729297	0	0	0																								
732323	0	0	0																								
737551	13	13	0	0.5	1	1	1.5	3	1	1.5	1.5	1.5	0.5		R1: Remove imagesize.y / imagesize.x from the x coordinate; R4: Shade_back always false, should be taken from args_. Second condition in material.cpp on row 29 should have < 0;												
765714	0	0	0																								
765756	11.5	11.5	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	0.5	0		R4: specular added even if light below local horizon. Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal; R8: mirrorDirection() not implemented.												
765785	17	14.5	2.5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	1.5		R9: Regular sampler and jittered sampler work only when m_dim == 3; Box: No parser support, i.e. boxes cannot be drawn, and intersection does not work correctly; Fresnel: Incorrect formula. Check wikipedia for Shlick's approximation;			2				0.5				
765882	3	3	0	0.5	1		1.5								R1: normalizedImageCoordinateFromPixelCoordinate flips x and y, and orthographic camera size is the size of the whole area so you should divide the offset from center by two.												
766108	0	0	0																								
767136	0	0	0																								
769396	0	0	0																								
772419	0	0	0																								
784465	0	0	0																								
784847	0	0	0																								
784902	0	0	0																								
785053	0	0	0																								
785134	17.5	14	3.5	0.5	1.5	1	1.5	3	1	1.5	1	1.5	1.5		R4: When shade_back is true, diffuse lighting is added twice and the first time the normal is in the wrong direction; R7: Parameter tmin for intersect should be 0.0, not tmin. Shadows with orthographic camera look incorrect because of this; R9: You are applying the same amount of jitter to both x and y coordinates, instead you should add separate random values for each axis. More primitives: Transform only; Refraction: Total internal reflection is readily handled by the reflection code path (so there shouldn't be an else in if(trans))	2		1.5								
785163	0	0	0																								
785228	23	14.5	8.5	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	2		R4: shade_back is a cmd line argument. The condition inside the if(shade_back) should have >0, not < 0. spec_dp should be 0 when light is below local horizon. Box: When deriving the normal, you should take the absolute value of tmp_V - min/max and FLT_EPSILON is too small a value; Refraction: Total internal reflection is readily handled by the reflection code path	2		2.5							4	Spectral representation (4p)
785257	0	0	0																								
785325	0	0	0																								
785354	18	14.5	3.5	0.5	1.5	1	1.5	3	1	1.5	1	1.5	2		R7: should not add old tmin to the epsilon; this will be -FLT_MAX for ortographic cameras! The way you wrote your refraction breaks reflective-only materials (see the r8_reflective_sphere scene).	1.5		2								
785367	18.5	14.5	4	0.5	1.5	1	1.5	3	1	1.5	1	1.5	2		R7: should not add previous tmin to shadow epsilon (for ortho cameras this is -FLT_MAX!)	2		2								
785435	0	0	0																								
785448	0	0	0												Code doesn't compile. You need to submit working programs, not snippets of code copy pasted from somewhere else.												
785451	0	0	0																								
785493	0	0.5	0	0.5										-0.5	README has not been filled out; Compiler errors (-0.5p);												
785503	0	0	0																								
785516	0	0	0																								
795551	8.5	8.5	0	0.5	1.5	1	1.5		1	1.5		1.5			The reflections look a bit odd without the other shading effects, but seem to be correct.												
795577	9	9.5	0	0	0.5	1	1.5	1.5	1		1	1.5	1.5	-0.5	Compiler errors (-0.5p); R0: The green component should depend on j and the blue component should be 1. Also, integer division means that values are 0 for almost all pixels; R1: Ambient lighting constant, normalized image coordinate vector has x coordinate in both indices; R4: shade_back not used. d and dot_pro not defined. Conditions in shade should have >0, not < 0. You shouldn't add anything when light is below local horizon.; R7: r should have dir_to_light as direction, not -dir_to_light; R9: Regular sampler does not return a sample through the nth subpixel. Jittered sampler not random when m_dim == 1.												
795593	0	0	0																								
795629	0	0	0																								
795658	0	0	0																								
795674	18	14.5	3.5	0.5	1.5	1	1.5	3	1	1.5	1	1.5	2		R7: shadowTmin should be directly set to a small value (tmin will be -FLT_MAX for orthographic cameras!) In the Ashikhmin-Shirley material, k2 should just be -ray.direction; the direction to the viewer.			2							1.5	Ashikhmin-Shirley BRDF
795713	0	0	0																								
795865	6	6	0	0.5	1.5	1	1	2							R1: To get the exact same images as the example, you should remove -1 from imageSize.x and imageSize.y; R3: dir not normalized; R4: dir_to_light not normalized in point light. The condition on row 28 in shade should have hitNormal, not dir_to_light. Specular term shouldn't be multiplied by diffuse color. Specular added even if light below local horizon;												
796178	14	14	0	0.5	1.5	1	1.5	2	1	1.5	1.5	1.5	2		R4: Specular should be 0 when the light is below the local horizon; Point light dir_to_light not normalized, shading broken for point lights;												
798257	0	0	0																								
801131	0	0	0																								
804646	14.5	14.5	0	0.5	1	1	1.5	3	1	1.5	1.5	1.5	2		R1: There should be no max_length, x coordinate should be divided by imageSize[0] and y coordinate with imageSize[1]												
807711	0	0	0																								
809609	4.5	4.5	0	0.5	1	1	1.5	0.5							R1: camera scaling off by factor of 2, R4: diffuse formula almost there.												
811383	0	0	0																								
814872	0	0	0																								
818315	0	0	0																								
821289	0	0	0																								
822709	0	0	0																								
46596K	0	0	0																								
55055P	0	0	0																								
62727K	18.5	15	3.5	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		More primitives: Transform only	2		1.5								
64879R	16	15	1	0.5	1.5	1	1.5	3	1	1.5	1.5	1.5	2		R4: Shade_back is about the direction of the incoming ray (not the light direction) with relation to the normal; More primitives: In transform, you should use inverse_, not matrix_, when calculating r0 and rd. No parsing for boxes so they cannot be drawn. In box intersect, no values assigned to h when function returns true. No normals calculated. Other mistakes as well in both.			1								
65451T	0	0	0																								
67932J	2.5	2.5	0	0.5	1	1									R1: y-coord flipped, origin incorrect (should depend on horizontal and up vectors) and thus only work for certain orientations.												
69246M	14	14	0	0.5	1.5	1	1.5	2.5	1	1.5	1.5	1.5	1.5		R4: Directional light distance incorrectly set to hit.t. This causes the issues with shadows since you're using that distance as the maximum length for shadow rays. Set the directional light distance instead to something large such as FLT_MAX.												

Student number	point total	req total	extra total	R0 UV (0.5p)	R1 ortho, ambient (1.5p)	R2 Depth vis (1p)	R3 Perspective (1.5p)	R4 Phong, lights (3p)	R5 Planes (1p)	R6 Triangles (1.5p)	R7 Shadows (1.5p)	R8 Reflection (1.5p)	R9 AA (2p)	mod	notes / extras / ...	Refraction (1-2p)	Simple fog (1p)	More primitives (3p)	Arbitrary filters (1-3p)	Stereo cubemap (4p+)	transparent shadows (1.5p)	Fresnel (1p)	Textures (2-4p)	Normal mapping (2-3p)	CSG (4-5p)	Other extras (?p)	What other extras
77241H	0	0	0																								
77388B	0	0	0																								
83107B	0	0	0																								
83854J	0	0	0																								
84171B	3.5	3.5	0	0.5	1	1	1								R1: Objects too small in images. Offset from center should be divided by two; R3: fov_angle already in radians, dir not normalized;												
84805K	0	0	0																								
k28342	0	0	0																								
k90624	0	0	0																								
k93517	0	0	0																								