CS5431 - Assignment 2 **Due: 28/02/2025**
Catmull-Rom Splines

Problem:

Write a program, named CG_hw2, which evaluates a 3D Catmull-Rom spline and approximates it with a polyline.
Your program will read in an arbitrary number of 3D points, along with the tangents at the first and last points, and will fit a Catmull-Rom spline to them.
A Catmull-Rom spline is a C1 piecewise curve that consists of concatenated cubic Bezier curves.
Each individual Bezier curve is parameterized from 0 to 1. The curves' *u* parameters will be incremented by *du* (*1/n*) during evaluation.
n is the number of line segments in the individual Bezier curves.
You should also add the "tension" property to your spline, as defined by Kochanek and Bartels.

Specification:

1. The program reads from a file two tangent vectors and the 3D points (3 floating point numbers) that will be interpolated by the spline, specified by the -f *filename* argument.
   The file will contain two tangent vectors followed by *N* 3D points, one per line, for example:
   dx1 dy1 dz1
   dx2 dy2 dz2
   x1 y1 z1
   x2 y2 z2
   x3 y3 z3
   x4 y4 z4
   ...
   Default value: cpts_in.txt
2. The *u* increment (*du*, a real number between 0 and 1) can be computed from the number of segments specified by the -n *n* argument. *du* = *1/n*. Default value: 11
3. The tension value (a real number less than 1) is specified by the -t *tension* argument. Default value: 0
4. The locations of the input points should be displayed with small spheres.
5. The radius of the spheres is specified by the -r *radius* argument. Default value: 0.1
6. Each cubic Bezier curve is parameterized from 0 to 1.
7. Be sure to evaluate the individual curves only in the range $0 \le u \le 1$, and that your spline interpolates all input points.
8. Write the resulting graphics primitives in the Open Inventor format to standard out.
9. Your program will be tested with the command like "./CG_hw2 -f *filename* -n *n* -r *radius* -t *tension* > out.iv"
10. Your program should not require arguments and should be able to process a subset of them in arbitrary order.

Example Output:

The first two examples are the same two from HW1, but in Catmull-Rom (Hermite) form.

i. The following input data
   4 0 4
   4 0 -4
   0 0 0
   5 0 0
   should produce this Catmull-Rom spline with
   n = 40 (du = 0.025), radius = 0.05, tension = 0 and this input file.
   should produce this Catmull-Rom spline with
   n = 40 (du = 0.025), radius = 0.05, tension = 0.5.
   should produce this Catmull-Rom spline with
   n = 40 (du = 0.025), radius = 0.05, tension = -0.5.

ii. The following input data

   4 5 1
   3 1 2
   0 0 0
   5 0 0
   should produce this Catmull-Rom spline with
   n = 40 (du = 0.025), radius = 0.05, tension = 0 and this input file.

iii. The following input data
   3 -7 1
   2 3 -6
   -3.5 5.1 1.2
   -1.1 3.2 2.8
   0.5 7.3 3.5
   4.7 6.9 2.2
   should produce this Catmull-Rom spline with
   n = 33 (du = 0.0303), radius = 0.1, tension = 0 and this input file.

iv. The following input data
   3 -7 1
   6 9 -18
   -3.5 5.1 1.2
   -1.1 3.2 2.8
   0.5 7.3 3.5
   4.7 6.9 2.2
   5.5 -1.0 -3.8
   should produce this Catmull-Rom spline with
   n = 25 (du = 0.04), radius = 0.1, tension = 0 and this input file.

v. The following input data
   3 -7 1
   6 9 -18
   -3.5 5.1 1.2
   1.1 3.2 -2.8
   0.5 7.3 3.5
   4.7 6.9 2.2
   6.5 1.0 -3.8

3.9 4.7 -1.3
6.9 -0.4 4.6
7.8 -4.2 5.6
9.6 2.7 -1.6

should produce this Catmull-Rom spline with
n = 50 (du = 0.02), radius = 0.075, tension = 0 and this input file.

vi. The following input data

7 1 5
-6 3 -10
-3.5 5.1 1.2
1.1 3.2 -2.8
0.5 7.3 3.5
4.7 6.9 2.2
6.5 1.0 -3.8
3.9 4.7 -1.3
6.9 -0.4 4.6
7.8 -4.2 5.6
9.6 2.7 -1.6

should produce this Catmull-Rom spline with
n = 50 (du = 0.02), radius = 0.075, tension = 0 and this input file.
should produce this Catmull-Rom spline with
n = 50 (du = 0.02), radius = 0.075, tension = 0.8.
should produce this Catmull-Rom spline with
n = 50 (du = 0.02), radius = 0.075, tension = -2.

Grading Scheme

1. Parsing input file correctly : 1 point
2. Correctly display input points with spheres : 1 point
3. Incrementing curve segments correctly : 1 point
4. Implement tension correctly : 1 point
5. Catmull-Rom spline evaluation correct : 6 points

Submission Guidelines:

1. Assignments must be submitted via thuto.
2. README file: explain the features of your program, language and OS used, compiler or interpreter used, name of file containing main(). Text files only.
3. All source code. Your code must compile and run on Linux.
4. You may program in any language you like as long it can produce a usable executable on Linux.
5. Your program will be run by the grader. Please do NOT submit any image files, Visual C++ project files, or anything not requested in this section. Your program must run on Linux without the installation of "special" libraries.

6. If you are using a language that doesn't produce an executable file, e.g. python, then be sure to include a script called CG_hw2 that accepts arguments and prints Inventor to standard out.
7. Points will be deducted if submission guidelines are not followed.