

ASSIGNMENT 1: Data Structures

[30 marks]

You have been provided with a simple reader for polygon soup. It has code for reading in a geometric object and rendering it on screen. Rendering is not actually necessary for this task, but it has been provided to make your task easier.

You have also been provided with a set of sample triangle soup files with the suffix ".tri", where the first line is the number of triangles (n), followed by 3n triplets giving the coordinates of each vertex.

You should:

- a) Write a program called 'face2faceindex' that reads in a .tri file, computes the face index format and outputs a file with ".face" suffix in the format on page 2. **[6 marks]**
- b) Write a second program called 'faceindex2directededge' that reads in a .face file, computes the directed edges and outputs a file with ".diredge" suffix in the format on page 3. **[6 marks]**
- c) Determine for each file provided whether it is manifold or not. If it is not, print out the ID number of the edge or vertex that fails. For this assignment, you need not worry about geometric self-intersection. **[6 marks]**
- d) For each mesh that you have demonstrated to be manifold, use the Euler formula to determine the genus of the surface. **[6 marks]**
- e) Analyse the algorithmic complexity of your code. **[6 marks]**

Note that you are expected to produce command-line utilities rather than a visual application. This is because, in the real world, you will need to convert file formats once rather than every time you load. If you wish, you may also modify the code handed out to read in a .face or .diredge file and render it directly to cross-check, but this is not required.

It is recommended that you start with the smallest files provided and work your way up in size. The .diredge file shown should not be used for testing, as it assumes that the face vertices are "edgeTo", not "edgeFrom", and will therefore not be the same as the ones you compute.

When the assignments are marked, at least one data file other than the ones provided will be used for testing. Make sure your code does not make any unnecessary assumptions.

All code should compile on the School's Linux machines without installation of any extra libraries or applications. You should include a makefile and a readme.txt file with any additional instructions.

PENALTIES:

Poorly structured or badly commented code may be penalised by up to 25% of the marks available. Poorly formatted output files may also be penalised by up to a further 25% of the marks available. Code without a readme or makefile may be penalised by up to 10% of the marks available. Code that does not compile properly will be assigned a mark of 0, but I will usually give the student one chance to correct this.

DUE DATE: Friday, November 11, 2022, 10:00am

APPENDIX 1: .face File Format

This file format begins with a header block with identifying information, and basic information about the file. It is similar in nature to the standard .obj file, but much simpler, as it leaves out normals, texture coordinates, &c.

You should have a header and two blocks in the following order:

1. Vertex block
2. Face block

In the vertex block, each vertex is listed by number with its xyz coordinates on a single line.

In the face block, each face is listed by number with its face indices on a single line in CCW order.

```
# University of Leeds 2022-2023
# COMP 5812M Assignment 1
# Your Name Goes Here
# Your Student Number Goes Here
#
# Object Name: Cube
# Vertices=8 Faces=12
#
Vertex 0  0.000000  0.000000  0.000000
Vertex 1  1.000000  0.000000  0.000000
Vertex 2  1.000000  0.000000  1.000000
Vertex 3  0.000000  0.000000  1.000000
Vertex 4  0.000000  1.000000  0.000000
Vertex 5  1.000000  1.000000  0.000000
Vertex 6  0.000000  1.000000  1.000000
Vertex 7  1.000000  1.000000  1.000000
Face  0  0  1  2
Face  1  0  2  3
Face  2  0  4  5
Face  3  0  5  1
Face  4  0  3  6
Face  5  0  6  4
Face  6  1  5  2
Face  7  5  7  2
Face  8  3  2  6
Face  9  2  7  6
Face 10  6  7  5
Face 11  6  5  4
```

APPENDIX 2: .diredge File Format

This file format is an extended version of the .face file format. It begins with the same header block, and adds additional blocks to the file.

You should have a header and four blocks in the following order:

1. Vertex block (as for the .face file)
2. First Directed Edge block
3. Face block (as for the .face file)
4. Other Half block

The reason why we insert 2. before 3. rather than after it is that the Vertex & FirstDirectedEdge blocks are the same size.

In the vertex block, each vertex is listed by number with its xyz coordinates on a single line.

In the first directed edge block, the ID of the vertex is listed, with the ID of the first directed edge at each vertex. Note that this is not canonical, as any directed edge from the vertex may be chosen.

In the face block, each face is listed by number with its face indices on a single line in CCW order.

In the other half block, each directed edge is listed by number, with the other directed edge pointing in the opposite direction that pairs with it

```
# University of Leeds 2022-2023
# COMP 5812M Assignment 1
# Your Name Goes Here
# Your Student Number Goes Here
#
# Object Name: Cube
# Vertices=8 Faces=12
#
Vertex 0  0.000000  0.000000  0.000000
Vertex 1  1.000000  0.000000  0.000000
Vertex 2  1.000000  0.000000  1.000000
Vertex 3  0.000000  0.000000  1.000000
Vertex 4  0.000000  1.000000  0.000000
Vertex 5  1.000000  1.000000  0.000000
Vertex 6  0.000000  1.000000  1.000000
Vertex 7  1.000000  1.000000  1.000000
FirstDirectedEdge 0  1
FirstDirectedEdge 1  2
FirstDirectedEdge 2  0
FirstDirectedEdge 3  3
FirstDirectedEdge 4  8
FirstDirectedEdge 5  6
FirstDirectedEdge 6  12
FirstDirectedEdge 7  23
Face 0  0  1  2
Face 1  0  2  3
Face 2  0  4  5
Face 3  0  5  1
Face 4  0  3  6
Face 5  0  6  4
```

Face	6	1	5	2
Face	7	5	7	2
Face	8	3	2	6
Face	9	2	7	6
Face	10	6	7	5
Face	11	6	5	4
OtherHalf	0		4	
OtherHalf	1		9	
OtherHalf	2		18	
OtherHalf	3		13	
OtherHalf	4		0	
OtherHalf	5		25	
OtherHalf	6		10	
OtherHalf	7		15	
OtherHalf	8		35	
OtherHalf	9		1	
OtherHalf	10		6	
OtherHalf	11		19	
OtherHalf	12		16	
OtherHalf	13		3	
OtherHalf	14		24	
OtherHalf	15		7	
OtherHalf	16		12	
OtherHalf	17		33	
OtherHalf	18		2	
OtherHalf	19		11	
OtherHalf	20		21	
OtherHalf	21		20	
OtherHalf	22		32	
OtherHalf	23		28	
OtherHalf	24		14	
OtherHalf	25		5	
OtherHalf	26		27	
OtherHalf	27		26	
OtherHalf	28		23	
OtherHalf	29		31	
OtherHalf	30		34	
OtherHalf	31		29	
OtherHalf	32		22	
OtherHalf	33		17	
OtherHalf	34		30	
OtherHalf	35		8	