Project 1, due on 02/14.

Problem. Use the Demo_Pkg code to implement the add first algorithm, which is to add a new node as the first node of the list. See Figure 1. After insertion, the new node is the "first" node in the list.

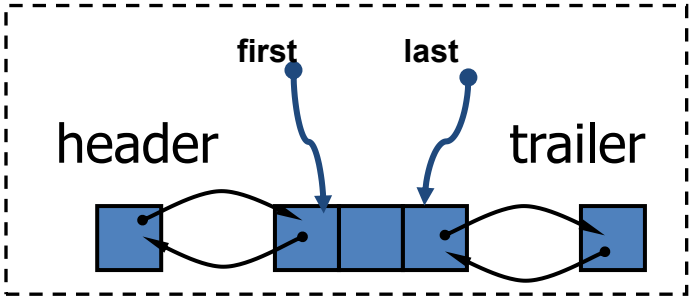

Figure 1.

The current code implements the add last algorithm in function insert_tri_at_tail_of_list(), which is to add a new node as the last node of the list.

Instructions:

1. Use the following declaration of the function to implement the add first algorithm:

LOCAL void insert_tri_at_head_of_list(TRI *tri, SURFACE *s);

Variable "tri" is the node to be inserted into the list maintained in "s".

2. Replace the insert_tri_at_tail_of_list() function by your insert_tri_at_head_of_list() function. In the main() function, use the following loop statement to print out ID of each of triangles in the list.

```
for (temp_tri = first_tri(outs); !at_end_of_tri_list(temp_tri,outs);
        temp_tri = temp_tri->next)
{
   printf("temp_tri ID = %d\n", temp_tri->id);
}
```

3. Construct the adjacency of triangles by installing to Tri_on_side() of each of the triangles, the edge adjacent triangles. Compare your results with the adjacency information in newBurgers.e file.

Hint: this adjacency information can be retrieved from "neigh" argument of the read_input_easy_mesh().

Hand-In. Make a tar ball of all of your source code. Turn in the report which contains results and pseudo-code description of your implementations. Email the tar ball of source code. Use the following title for your email: acms40212S14-Proj1-your-ND-ID