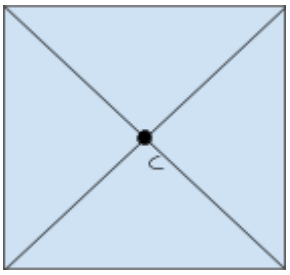
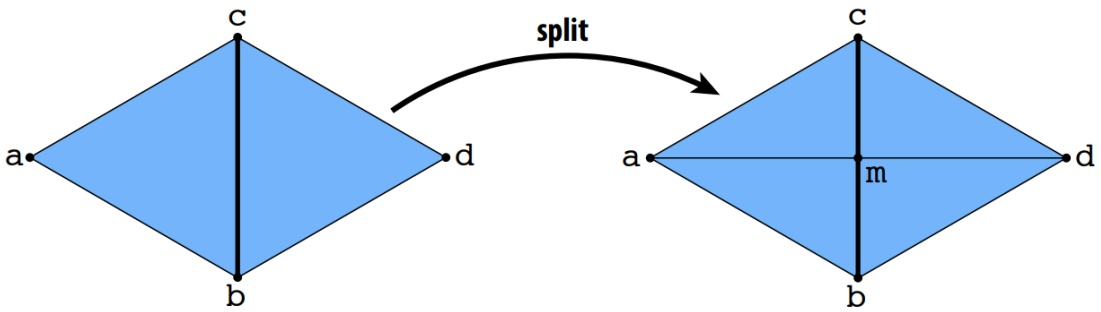
Paul Mallard & Pauline Dos Santos E4FI group 2

Question 3 : Given a **myMesh** M, and a **myFace** f, give the pseudo-code to split f by adding its centroid c as a new vertex in M. f will then be broken into many triangles, each triangle formed by c and two consecutive vertices on the boundary of f. Note that f need not be a triangle, and can have many vertices on its boundary.






```
void splitF(myMesh* M, myFace* f){
    std::vector<myVertex*> v;
    myHalfedge* start = f->halfedge;
    myHalfedge* e = start;

    do {
        v.push_back(e->vertex);
        e = e->next;
    } while (e != start);

//centroid
    Point3D c(0, 0, 0);
    for (myVertex* n : v) {
        c += n->position;
    }
    c /= v.size();

    myVertex* center= M->addVertex(c);

    std::vector<myHalfedge*> newHalfedges;

//triangulation
```

```
    for (size_t i = 0; i < v.size(); ++i) {
        myVertex* n1 = v[i];
        myVertex* n2 = v[(i + 1) % v.size()];

        myFace* newF = M->addFace();

        myHalfedge* e1 = M->addHalfedge();
        myHalfedge* e2 = M->addHalfedge();
        myHalfedge* e3 = M->addHalfedge();


        e1->vertex = n1;
        e2->vertex = n2;
        e3->vertex = center;

        e1->next = e2; e2->next = e3; e3->next = e1;
        e1->prev = e3; e2->prev = e1; e3->prev = e2;

        e1->adjacent_face = newF;
        e2->adjacent_face = newF;
        e3->adjacent_face = newF;
        newF->halfedge = e1;

        newHalfedges.push_back(e1);
        newHalfedges.push_back(e2);
        newHalfedges.push_back(e3);
    }

//Associate twins and halfedges
    for (size_t i = 0; i < newHalfedges.size(); ++i){
        myHalfedge* e1 = newHalfedges[i];
        for (size_t j = i + 1; j < newHalfedges.size(); ++j){
            myHalfedge* e2 = newHalfedges[j];
            if (e1->vertex == e2->next->vertex && e2->vertex == e1->next->vertex) {
                e1->twin = e2;
                e2->twin = e1;
            }
        }
    }

    M->deleteFace(f);
}
```