Individual test. Do not look at other students’ work. Please type and write legibly. Bring to class. All 2D.

1) Write the code for `EE(pt A, pt B, pt P, pt Q) {...}`, which returns `true` when edge(A,B) and edge(P,Q) intersect. If your code uses other functions (except for `dot()` and other trivial point and vector operators), please provide the code for them as well. (You should test your code in Processing before including it here.)

```java
boolean EE(pt A, pt B, pt P, pt Q) {boolean hit=true;
if (isLeftTurn(A,B,C)==isLeftTurn(A,B,D)) hit=false;
if (isLeftTurn(C,D,A)==isLeftTurn(C,D,B)) hit=false; return hit; }
```

2) Write the code for `EC(pt A, pt B, pt C, float r) {...}`, which, if edge(A,B) does not intersect circle(C,r) returns -1, and otherwise returns the value of the parameter t of the point X=A+tAB which is the first intersection where the ray from A to B hits the circle. If your code uses other functions (except for `dot()` and other trivial point and vector operators), please provide the code for them as well. (You should test your code in Processing.)

```java
float EC (pt A, pt B, pt C, float r) {
vec T = V(A,B); float n = n(T); T.normalize(); vec AC = V(A,C);
float d = dot(AC,T); float h = dot(AC,R(T)); float t = -1;
if (h<r) { float w = sqrt(sq(r)-sq(h)), t1 = (d-w)/n, t2 = (d+w)/n;
if ((0<=t1)&&(t1<=1)) t = t1; else if ((0<=t2)&&(t2<=1)) t = t2; }
return t; }
```

3) Consider a control polygon P. Explain the 4-point subdivision technique. Assume that consecutive vertices at one subdivision levels are named A, B, C, D…. Explain how you obtain the new vertices B₁ and B₂ corresponding to B and the edge BC, using the linear interpolation function s(P,t,Q). Point out the advantages and limitations of this scheme.

```java
B₁ = B; B₂ = S(S(A,9/8,B),1/2,S(D,9/8,C)). Interpolating, but only C¹.
```

4) Consider a control polygon P. Explain the cubic B-spline subdivision technique. Assume that consecutive vertices at one subdivision levels are named A, B, C, D…. Explain how you obtain the new vertices B₁ and B₂ corresponding to B and the edge BC, using the linear interpolation function s(P,t,Q). Point out the advantages and limitations of this scheme.

```java
B₁ = S(S(B,1/4,A),1/2,S(B,1/4,C)); B₂ = S(B,1/2, C). C², but not interpolating.
Span(BC) lies in the convex hull of (A,B,C,D), which is useful for collision and clipping.
```

5) Suggest a good approximation of the velocity (tangent vector) V at point B in a sequence ...A,B,C.... of a polyloop:

```java
V=S(0.5,V(A,C)); // V=AC/2;
```