POINTS AND VECTORS (30/30 points)

The given prelude defines three types, one for three dimensional points, another for velocity vectors in three dimensions, and another one representing moving objects in space.

1. Write a function `move : point -> dpoint -> point` such that `move p dp` is the point `p` whose coordinates have been updated according to `dp`.
   (`x` is now `x +. dx` , `y` is now `y +. dy` , `z` is now `z +. dz` .
2. Write a function `next : physical_object -> physical_object` such that `next o` is the physical object `o` at time `t + dt` .
   The position of `next o` is the position of `o` moved according to its velocity vector.
3. Suppose that these objects are spheres whose radius is `1.0` .
   Write a function `will_collide_soon : physical_object -> physical_object -> bool` that tells if at the next instant, the two spheres will intersect.

THE GIVEN PRELUDE

```
type point  = { x : float; y : float; z : float }
type dpoint = { dx : float; dy : float; dz : float }
type physical_object = { position : point; velocity : dpoint }
```

YOUR OCAML ENVIRONMENT

```
1  let move p dp =
2    {x= p.x +. dp.dx ; y = p.y +. dp.dy ; z = p.z +. dp.dz} ;;
3
4  let next obj =
5    {position = move obj.position obj.velocity;velocity=obj.velocity};;
6
7
8  let will_collide_soon p1 p2 =
9    if sqrt (
10       ((next p1).position.x -. (next p2).position.x) *.
11       ((next p1).position.x -. (next p2).position.x)
12       +.
13       ((next p1).position.y -. (next p2).position.y) *.
14       ((next p1).position.y -. (next p2).position.y)
15       +.
16       ((next p1).position.z -. (next p2).position.z) *.
17       ((next p1).position.z -. (next p2).position.z)
18     ) < 2.0
19   then true else false;;
20
```

Evaluate >

Switch >>

Typecheck

Reset Templ

Full-screen

Check & Sa

Exercise complete (click for details) 30 pts

v Exercise 1: `move` Completed, 10 pts

Found `move` with compatible type.

Computing
`move {x = 0.864; y = -1.942; z = 1.526} {dx = 0.411; dy = 0.551; dz = 0.676}`
Correct value `{x = 1.275; y = -1.391; z = 2.202}` 1 pt

Computing
`move`
`{x = -1.424; y = 1.604; z = -0.343}`
`{dx = -0.244; dy = -0.064; dz = 0.414}`
Correct value `{x = -1.667; y = 1.540; z = 0.071}` 1 pt

Computing
`move`
`{x = 2.419; y = -1.850; z = -1.120}`
`{dx = 0.209; dy = 0.850; dz = -0.723}`
Correct value `{x = 2.627; y = -1.000; z = -1.843}` 1 pt

Computing
`move`
`{x = -0.377; y = 0.674; z = 0.729}`
`{dx = -0.222; dy = -0.099; dz = -0.026}`

Correct value {x = 3.148; y = 0.275; z = -1.067} 1 pt

Computing
move
{x = 2.084; y = -2.241; z = 0.663}
{dx = 0.253; dy = -0.353; dz = -0.750}
Correct value {x = 2.337; y = -2.595; z = -0.087} 1 pt

Computing
move {x = 1.631; y = -2.319; z = -0.268} {dx = 0.165; dy = 0.541; dz = 0.470}
Correct value {x = 1.796; y = -1.779; z = 0.202} 1 pt

Computing
move
{x = -2.270; y = 0.835; z = -1.326}
{dx = -0.595; dy = 0.495; dz = -0.606}
Correct value {x = -2.864; y = 1.330; z = -1.932} 1 pt

Computing move {x = 2.093; y = 0.549; z = 1.345} {dx = 0.503; dy = 0.328; dz = 0.930}
Correct value {x = 2.596; y = 0.877; z = 2.275} 1 pt

Computing
move {x = 1.784; y = 0.619; z = -0.487} {dx = -0.401; dy = 0.329; dz = 0.236}
Correct value {x = 1.383; y = 0.948; z = -0.251} 1 pt

Exercise 2: next — Completed, 10 pts

Found next with compatible type.

Computing
next
{position = {x = 1.647; y = -2.146; z = -0.891};
velocity = {dx = 0.490; dy = 0.292; dz = -0.569}}
Correct value 1 pt
{position = {x = 2.137; y = -1.854; z = -1.460};
velocity = {dx = 0.490; dy = 0.292; dz = -0.569}}

Computing
next
{position = {x = -0.830; y = -0.329; z = 1.000};
velocity = {dx = 0.806; dy = -0.418; dz = 0.275}}
Correct value 1 pt
{position = {x = -0.024; y = -0.747; z = 1.274};
velocity = {dx = 0.806; dy = -0.418; dz = 0.275}}

Computing
next
{position = {x = 0.012; y = -0.440; z = 1.576};
velocity = {dx = -0.878; dy = 0.931; dz = 0.125}}
Correct value 1 pt
{position = {x = -0.866; y = 0.491; z = 1.700};
velocity = {dx = -0.878; dy = 0.931; dz = 0.125}}

Computing
next
{position = {x = -0.279; y = -0.149; z = 2.233};
velocity = {dx = 0.987; dy = 0.759; dz = -0.630}}
Correct value 1 pt
{position = {x = 0.707; y = 0.610; z = 1.603};
velocity = {dx = 0.987; dy = 0.759; dz = -0.630}}

Computing
next
{position = {x = -0.782; y = 0.687; z = -0.580};
velocity = {dx = 0.095; dy = 0.004; dz = 0.477}}
Correct value 1 pt
{position = {x = -0.687; y = 0.691; z = -0.103};
velocity = {dx = 0.095; dy = 0.004; dz = 0.477}}

Computing
next
{position = {x = 0.187; y = 1.201; z = -1.072};
velocity = {dx = -0.392; dy = -0.632; dz = 0.255}}
Correct value 1 pt
{position = {x = -0.205; y = 0.569; z = -0.817};
velocity = {dx = -0.392; dy = -0.632; dz = 0.255}}

Computing
next
{position = {x = 2.114; y = 2.287; z = -0.926};
velocity = {dx = -0.170; dy = 0.301; dz = 0.482}}
Correct value 1 pt
{position = {x = 1.944; y = 2.588; z = -0.444};
velocity = {dx = -0.170; dy = 0.301; dz = 0.482}}

Computing
next
{position = {x = -0.861; y = -2.145; z = -1.651};
velocity = {dx = 0.557; dy = 0.412; dz = -0.646}}
Correct value 1 pt
{position = {x = -0.304; y = -1.733; z = -2.297};
velocity = {dx = 0.557; dy = 0.412; dz = -0.646}}

Computing
next
{position = {x = 1.417; y = 0.655; z = -0.727};
velocity = {dx = -0.640; dy = -0.346; dz = 0.148}}
Correct value 1 pt

```
    {position = {x = 0.162; y = -0.819; z = 1.309};
     velocity = {dx = -0.843; dy = -0.843; dz = -0.644}}
```
Correct value 1 pt
```
  {position = {x = -0.681; y = -1.662; z = 0.665};
   velocity = {dx = -0.843; dy = -0.843; dz = -0.644}}
```

v Exercise 3: `will_collide_soon` Completed, 10 pts

Found `will_collide_soon` with compatible type.

Computing
```
  will_collide_soon
    {position = {x = -1.289; y = -1.911; z = -0.445};
     velocity = {dx = 0.145; dy = -0.159; dz = -0.165}}
    {position = {x = -0.250; y = -2.315; z = -1.941};
     velocity = {dx = 0.979; dy = 0.622; dz = -0.463}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = 1.949; y = -2.193; z = -0.004};
     velocity = {dx = -0.208; dy = 0.411; dz = -0.900}}
    {position = {x = 1.068; y = 1.677; z = -2.359};
     velocity = {dx = 0.053; dy = -0.886; dz = -0.695}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -1.839; y = -1.104; z = 2.361};
     velocity = {dx = 0.250; dy = 0.875; dz = -0.541}}
    {position = {x = -2.031; y = 0.510; z = 2.220};
     velocity = {dx = -0.839; dy = -0.527; dz = -0.553}}
```
Correct value `true` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -2.427; y = 1.321; z = 0.476};
     velocity = {dx = -0.849; dy = -0.591; dz = 0.687}}
    {position = {x = -0.140; y = 0.302; z = -1.581};
     velocity = {dx = -0.775; dy = 0.464; dz = -0.828}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -2.163; y = -0.622; z = -1.310};
     velocity = {dx = -0.950; dy = 0.584; dz = 0.427}}
    {position = {x = -0.436; y = 0.354; z = 1.654};
     velocity = {dx = -0.217; dy = 0.936; dz = -0.513}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -2.363; y = -0.460; z = 1.194};
     velocity = {dx = -0.436; dy = -0.350; dz = -0.171}}
    {position = {x = 1.908; y = 0.901; z = -1.247};
     velocity = {dx = -0.552; dy = 0.037; dz = 0.071}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -1.204; y = 0.298; z = 1.713};
     velocity = {dx = 0.523; dy = 0.406; dz = -0.031}}
    {position = {x = -0.608; y = -0.398; z = 0.233};
     velocity = {dx = -0.180; dy = -0.362; dz = 0.656}}
```
Correct value `true` 1 pt

Computing
```
  will_collide_soon
    {position = {x = 0.343; y = -1.606; z = 1.300};
     velocity = {dx = 0.125; dy = 0.427; dz = 0.763}}
    {position = {x = -0.270; y = 0.499; z = -0.968};
     velocity = {dx = -0.790; dy = 0.653; dz = -0.166}}
```
Correct value `false` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -2.035; y = 1.131; z = -1.137};
     velocity = {dx = 0.056; dy = 0.334; dz = 0.640}}
    {position = {x = -2.424; y = 0.171; z = -2.018};
     velocity = {dx = -0.845; dy = 0.028; dz = 0.813}}
```
Correct value `true` 1 pt

Computing
```
  will_collide_soon
    {position = {x = -2.145; y = -2.169; z = -1.176};
     velocity = {dx = -0.337; dy = -0.581; dz = 0.861}}
    {position = {x = -2.316; y = -1.061; z = -0.802};
     velocity = {dx = -0.861; dy = -0.255; dz = -0.884}}
```
Correct value `false` 1 pt