```
[u,v,z] = multiaccess
```

```
u = 100×1
    -1
    -1
     1
     1
     1
    -1
     1
     1
    -1
    -1
     ⋮
v = 100×1
     1
     1
    -1
     1
    -1
     1
     1
     1
     1
     1
     ⋮
z = 500×1
   -1.8835
   -1.9373
    2.0075
    0.0352
    1.9303
   -1.8304
    0.0059
    0.1797
   -1.9736
   -1.9128
     ⋮
```

```
n = length(u);
m = length(z)/n;

%a) Calculate the angle between the code vectors u and v.

theta = acosd(dot(u,v)/(norm(u)*norm(v)));
fprintf('angle(u,v) = %.4f degrees\n', theta);
```

```
angle(u,v) = 92.2924 degrees
```

```
%How does this affect the decoding scheme? %Is it still possible to compute the
binary sequences b and c from z?
%Due to the angles being approximately orthogonal, each projection contains
%a small leakage, and with noise it makes the projection less exact,
```

```
%however since the angle is close to 90 degrees and the noise is small, the
%decoding scheme still works.

%Since the angle is near 90 degrees (92.2924) they are nearly orthogonal.

%b) Compute (b1,b2,...,b5) and (c1,c2,...,c5)
b_hat = zeros(m,1);
c_hat = zeros(m,1);

for k = 1:m
    zk = z((k-1) * n + (1:n));
    b_soft = dot(u, zk)/norm(u)^2;
    c_soft = dot(v, zk)/norm(v)^2;

    b_hat(k) = sign(b_soft);
    if b_hat(k) == 0
        b_hat(k) = 1;
    end

    c_hat(k) = sign(c_soft);
    if c_hat(k) == 0
        c_hat(k) = 1;
    end
end

disp('b (projection) =');
```

```
b (projection) =
```

```
disp(b_hat.')
```

```
     1    -1    -1     1    -1
```

```
disp('c (projection) =');
```

```
c (projection) =
```

```
disp(c_hat.')
```

```
    -1     1    -1    -1     1
```