```
program matrix;

 type
  colvector = array[1..3] of real;
  matrix = array[1..3, 1..3] of real;
  determinant = array[1..2,1..2] of real;
  matname = string[20];

 var
  det : real;
  current:integer;
  m,n,o,minv : matrix;
  v, xyz : colvector;

 procedure pause;
  var
   c : char;
 begin
  write('Press return key to continue.......');
  readln(c);
  writeln;
 end;

 procedure getvector (p:matname; var c:colvector);
  var
   temp : colvector;
   r : integer;
 begin
  writeln(p);
  writeln;
  for r := 1 to 3 do
   begin
    write('Enter vector element no.', r : 1, '       ');
    readln(temp[r])
   end;
  c:= temp
 end;

 procedure printname (n :matname);
 begin
  if length(n) = 0 then
   writeln('unnamed')
  else
   writeln(n)
 end;

 procedure printcolvector (s1, s2, s3 :matname ;
        c : colvector);

 begin
```

```
  writeln;
  writeln(s1, c[1] : 10 : 2);
  writeln(s2, c[2] : 10 : 2);
  writeln(s3, c[3] : 10 : 2);
  writeln;
 end;


 procedure printmatrix (title : matname;
         m : matrix);
  var
   r, c : integer;
 begin
  writeln;
  writeln(title);
  for r := 1 to 3 do
   begin
    write('                   ');
    for c := 1 to 3 do
     write(m[r, c] : 10 : 2);
    writeln;
   end;
  writeln;
  writeln;
 end;

procedure minor(m:matrix;row,col:integer;var result:determinant);
var r,c:integer;
    dr,dc:integer;
    p:integer;

begin
     p:=2;
     for r:=1 to 3 do
     begin
         for c:=1 to 3 do
         begin
              if (r<>row) and (c<>col) then
              begin
                   p:=p+1;
                   dr:=p div 2;
                   dc:=p mod 2;
                   result[dr,dc]:=m[r,c]
              end;
          end;
     end;
end;

function evaldeterminant(d:determinant):real;
begin
```

```
     evaldeterminant:=d[1,1]*d[2,2]-d[1,2]*d[2,1]
end;


 procedure transpose (old : matrix; var new : matrix);
  var
   r, c : integer;
 begin
  for r := 1 to 3 do
   for c := 1 to 3 do
    new[c, r] := old[r, c];
 end;

 function determ3x3 (m : matrix) : real;
  var
   minors:array [1..3] of determinant;
   d:array[1..3] of real;
   i:integer;
   temp:real;

 begin
  for i:=1 to 3 do
      begin
      minor(m,1,i,minors[i]);
      d[i]:= evaldeterminant(minors[i]);
      end;
  temp:=0;
  for i:= 1 to 3 do
      begin
      if (odd(i)) then
         temp:=temp+m[1,i]*d[i]
         else
         temp:= temp-m[1,i]*d[i];
      end;
  determ3x3:=temp
 end;

procedure divmat (m : matrix;
         det : real; var result: matrix);
  var
   r, c : integer;
   temp : matrix;

 begin
 if det<>0 then
 begin
  for r := 1 to 3 do
   for c := 1 to 3 do
    temp[r, c] := m[r, c] / det;
  result := temp;
 end
 else
 writeln('Determinant is Zero. Unable to divide into matrix')
 end;

 procedure getmatrix (p : matname;
         var m : matrix);
  var
   r, c : integer;
 begin
  ClrScr;
  writeln(p);
  writeln;
  for r := 1 to 3 do
   for c := 1 to 3 do
    begin
     write('Enter Row', r : 1, '  Column ', c : 1, '                  ');
     readln(m[r, c])
    end;
  writeln;
  pause;
  ClrScr;
 end;

 procedure getminor (m : matrix; var result:matrix);
  var d:determinant;
      r,c:integer;
begin
  for r:= 1 to 3 do
      for c:= 1 to 3 do
      begin
           minor(m,r,c,d);
           result[r,c]:=evaldeterminant(d)
      end;
 end;


 procedure cofactor (var m,n : matrix);
  var
   r, c : integer;

 begin
      for r:= 1 to 3 do
          for c:= 1 to 3 do
          begin
               if (odd(r+c)) then n[r,c]:=-1*m[r,c]
               else
               n[r,c]:=m[r,c]
          end;
 end;
```

```
procedure mult3x3 (var f,s,q : matrix) ;
 var
  r, c, p : integer;
  sum : real;
  temp : matrix;
begin
 for r := 1 to 3 do
  for c := 1 to 3 do
   begin
    sum := 0;
    for p := 1 to 3 do
     sum := sum + f[r, p] * s[p, c];
    temp[r, c] := sum
   end;
 q := temp;
end;

procedure matxcolvec (m : matrix;
        c : colvector;var result:colvector);
 var
  r : integer;
  temp : colvector;
begin
 for r := 1 to 3 do
  temp[r] := m[r, 1] * c[1] + m[r, 2] * c[2] + m[r, 3] * c[3];
 result := temp
end;


procedure getinverse (m : matrix;
        var inverse : matrix);
 var
  det : real;
  tmp1,tmp2,tmp3: matrix;
begin
 printmatrix('A', m);
 getminor(m,tmp1);
 printmatrix('Minor', tmp1);
 pause;
 cofactor(tmp1,tmp2);
 printmatrix('Cofactor', tmp2);
 pause;
 transpose(tmp2,tmp3);
 printmatrix('Transpose', tmp3);
 pause;
 det := determ3x3(m);
 writeln;
 writeln('Determinant is:', det : 10 : 2);
 if det<>0 then
```

```
  divmat(tmp3, det, inverse)
  else
  writeln('No inverse exists for this matrix')
 end;


 procedure menu (var decision : integer);
  var
   option, x : integer;
 begin
  ClrScr;
  writeln('                             Menu');
  writeln('                             ____ ');
  writeln;
  writeln('Inverse of a matrix                                    1');
  writeln('Determinant of a matrix                                2');
  writeln('Multiply two 3x3 Matrices                              3');
  writeln('Solve set of simultaneous Equations                    4');
  writeln('Multiply 3x3 matrix by vector                          5');
  writeln('Quit                                                   6');
  writeln;
  write('Enter required option...........');
  readln(option);
  if (option > 6) then
   menu(x)
  else
   decision := option;
 end;

procedure doit;
var choice:integer;
begin
 menu(choice);
 case choice of
  1 :
   begin
    getmatrix('Enter matrix.......',m);
    getinverse(m,minv);
    printmatrix('Original Matrix', m);
    printmatrix('Inverse', minv);
    pause;
   end;
  2 :
   begin
    getmatrix('Enter matrix.......', m);
    printmatrix('Matrix is: ', m);
    det := determ3x3(m);
    writeln('Determinant is ', det : 1 : 2);
    pause;
   end;
  3 :
   begin
    getmatrix('Enter first matrix.........', m);
    getmatrix('Enter second matrix.........', n);
    mult3x3(m,n,o);
    pause;
    printmatrix('First Matrix', m);
    printmatrix('Second Matrix', n);
    printmatrix('Product is:-', o);
    pause;
   end;
  4 :
   begin
    getmatrix('Enter matrix of coefficients....', m);
    getvector('Enter values of RHS of linear equations.....',v);
    getinverse(m,minv);
    matxcolvec(minv,v,xyz);
    printcolvector('X = ', 'Y = ', 'Z = ', xyz);
    pause;
   end;
  5 :
   begin
    getmatrix('Enter 3x3 matrix.....',m);
    getvector('Enter elements of column vector......',v);
    matxcolvec(m,v,xyz);
    printcolvector('1st element = ', '2nd element = ', '3rd element = ',
xyz);
    pause;
   end;
  6 :writeln('Bye......');
 end;
 if choice<>6 then doit
end;

begin
     doit
end.
```