```
//POWER & INVERSE POWER METHOD : Finding largest & smallest Eigenvalues

clc; clear; fd=%io(2); //File Descriptor(=6)

A=[2 -1 0; -1 2 -1; 0 -1 2]; //Input Data
X=[1; 1; 1]; //Initial Eigenvector
iter=1; //Set iteration number to 1
maxerr=1e-2;    //Set tolerance error (max)
err=1e+4;  //Set initial error (as large as possible)
lam1=1e10;   //Set initial Eigen value (as large as possible)

choice=int(input("Enter the choice: (1 - Largest & 2 - Smallest ) "));
if choice==1 then   //Finding largest Eigenvalue
    B=A;
elseif choice==2 then    //Finding smallest Eigenvalue
    B=inv(A);
else
    disp("Invalid Choice (Choose between 1 & 2)");
end

while(err>maxerr)
    xold=X; //Preserve old Eigenvector
    Y=B*X;  //Compute new matrix
    eigval=max(abs(Y)); //Compute EigenValue (largest)
    eigvec=Y./eigval;   //Compute new Eigenvector
    X=eigvec;   //Store Eigenvector values
    err=abs(sum(xold-X));   //Compute error
    lam1=eigval;    //Update EigenValue
    iter=iter+1;    //Update iteration counter
end

mfprintf(fd,"Method converge in %d iteration \n\n",iter-1);
if choice==1 then
    mfprintf(fd,"Greatest EigenValue = %5.5f\n",lam1);
elseif choice==2 then
    mfprintf(fd,"Smallest EigenValue = %5.5f\n",1/lam1);
end
disp("The corresponding eigenvector is:",X);

//OUTPUT No 1:
```

```
//Enter the choice: (1 - Largest & 2 - Smallest ) 1
//
//Method converge in 6 iteration
//
//Greatest EigenValue = 3.41667
//
//  "The corresponding eigenvector is:"
//
//   0.7073171
//  -1.
//   0.7073171
//Output No 2 :
//Enter the choice: (1 - Largest & 2 - Smallest ) 2
//
//Method converge in 4 iteration
//
//Smallest EigenValue = 0.58537
//
//  "The corresponding eigenvector is:"
//
//   0.7073171
//   1.
//   0.7073171
```