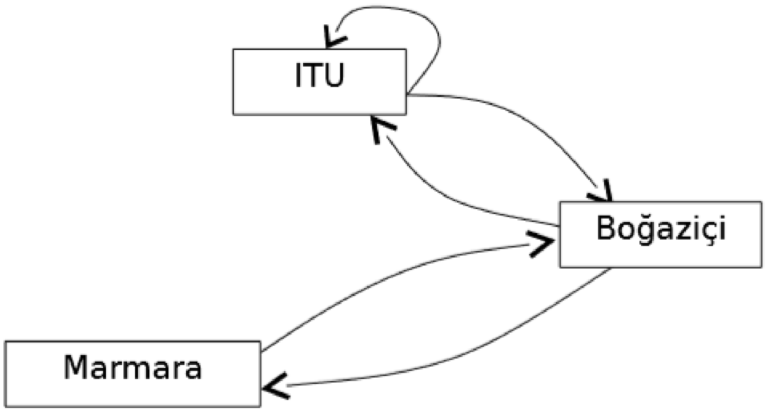
MATH 259 – Project #2

1)

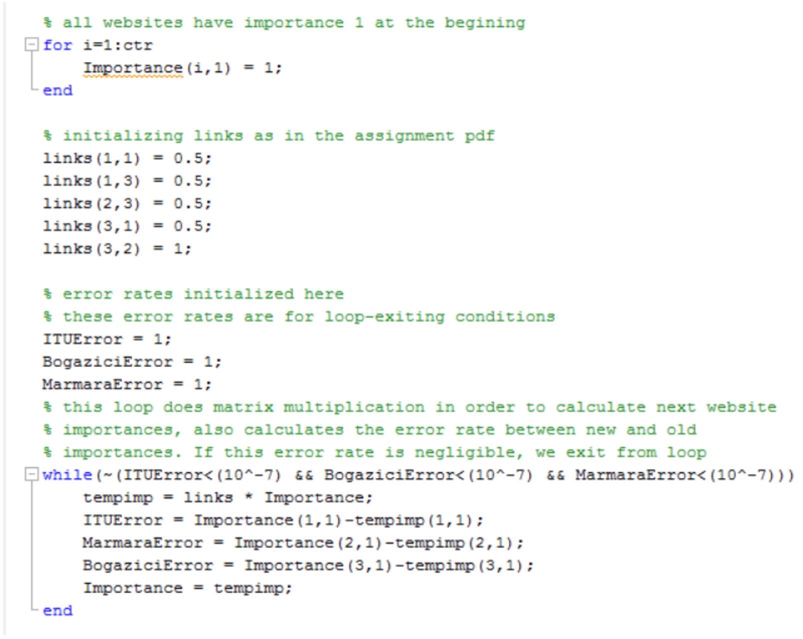


In this question, there are some websites of 3 schools, and some of them give links to each other or itself.

There is a matrix computation which calculates the importances of each school's importance. Initially they all have 1 importance.

$$\begin{bmatrix} I^{(t+1)} \\ M^{(t+1)} \\ B^{(t+1)} \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 \\ 1/2 & 1 & 0 \end{bmatrix} \begin{bmatrix} I^{(t)} \\ M^{(t)} \\ B^{(t)} \end{bmatrix}$$

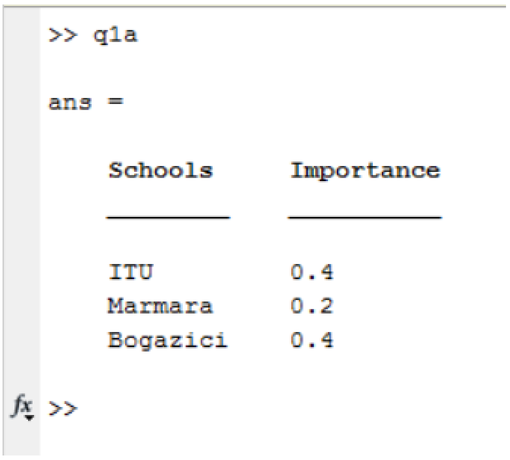
a- In this question, we are asked to find the last importance vector and normalize it. Marmara gives one link to Boğaziçi, Boğaziçi gives Marmara and ITU, ITU gives Boğaziçi and itself.



```
% all websites have importance 1 at the begining
for i=1:ctr
    Importance(i,1) = 1;
end

% initializing links as in the assignment pdf
links(1,1) = 0.5;
links(1,3) = 0.5;
links(2,3) = 0.5;
links(3,1) = 0.5;
links(3,2) = 1;

% error rates initialized here
% these error rates are for loop-exiting conditions
ITUError = 1;
BogaziciError = 1;
MarmaraError = 1;
% this loop does matrix multiplication in order to calculate next website
% importances, also calculates the error rate between new and old
% importances. If this error rate is negligible, we exit from loop
while(~(ITUError<(10^-7) && BogaziciError<(10^-7) && MarmaraError<(10^-7)))
    tempimp = links * Importance;
    ITUError = Importance(1,1)-tempimp(1,1);
    MarmaraError = Importance(2,1)-tempimp(2,1);
    BogaziciError = Importance(3,1)-tempimp(3,1);
    Importance = tempimp;
end
```

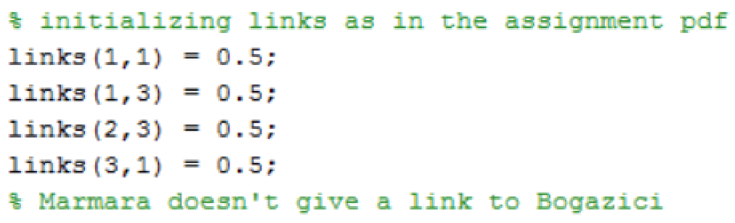
When absolute error is less than 0.0000001, loop stops and we get last importances. Also we normalize the final vector by summing all elements of the final vector, divide the elements of vector by it.

Sample output:



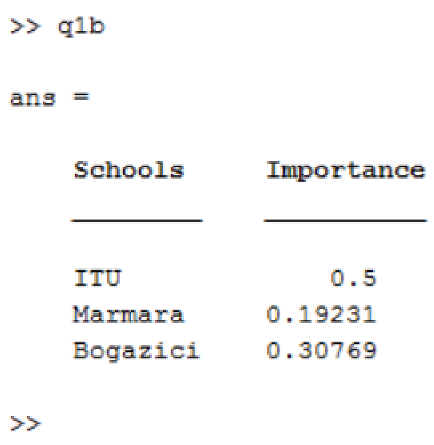
```
>> q1a

ans = 

    Schools       Importance
    _________    ___________

    ITU           0.4
    Marmara       0.2
    Bogazici      0.4

>> 
```

b- This question asks that what happens if Marmara doesn't give link to Boğaziçi. It means that element (3, 2) will be 0.



```
% initializing links as in the assignment pdf
links(1,1) = 0.5;
links(1,3) = 0.5;
links(2,3) = 0.5;
links(3,1) = 0.5;
% Marmara doesn't give a link to Bogazici
```

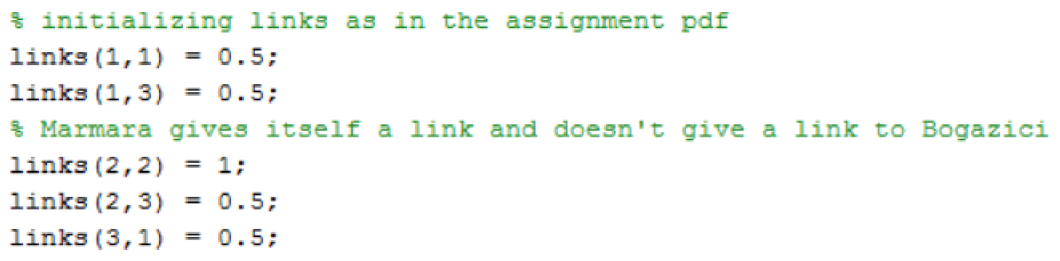
Sample output:



```
>> q1b

ans = 

    Schools       Importance
    _________    ___________

    ITU              0.5
    Marmara      0.19231
    Bogazici     0.30769

>> 
```

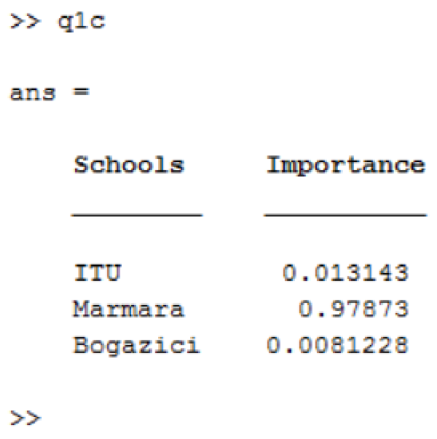
As we can see, Boğaziçi has less importance now.

c- If Marmara wants to be the most important website, Marmara should give itself a link, while it isn't giving a link to Boğaziçi. This will increase it's importance.



```
% initializing links as in the assignment pdf
links(1,1) = 0.5;
links(1,3) = 0.5;
% Marmara gives itself a link and doesn't give a link to Bogazici
links(2,2) = 1;
links(2,3) = 0.5;
links(3,1) = 0.5;
```

Sample output:



```
>> q1c

ans = 

    Schools       Importance
    _________    ___________

    ITU            0.013143
    Marmara         0.97873
    Bogazici      0.0081228

>>
```

2) In second question, we are asked to implement Gaussian Elimination with Scaled Partial Pivoting algorithm to solve an n x n matrix.

- Firstly we find elements which have the maximum absolute value from each row and call them s_i for i = 1, 2, …, n.

- After that, we divide all of the first column's elements with their rows' largest elements which is called s.

- The row which has the biggest ratio comes top of the matrix (swapped with the first row , also their ss' were swapped)

- Then we find the multipliers for second, third, … n th rows of the first column(m_{i1} for i = 2, 3, …, n). After that we apply Gaussian Elimination(e.g. $E_1 - (m_{21} * E_2)$) with finded multipliers in order to make all of the elements of the first column zero except for first element.

- After that, we eliminate first row and first column and we will have a matrix which is (n-1) x (n-1) sized. Then we apply the upper steps to our new (n-1) x (n-1) matrix. When we have an upper triangular matrix, we stop and start backward substition.

- Note that s isn't changing despite the basic row operations. Only swaps if their lines are swapped.

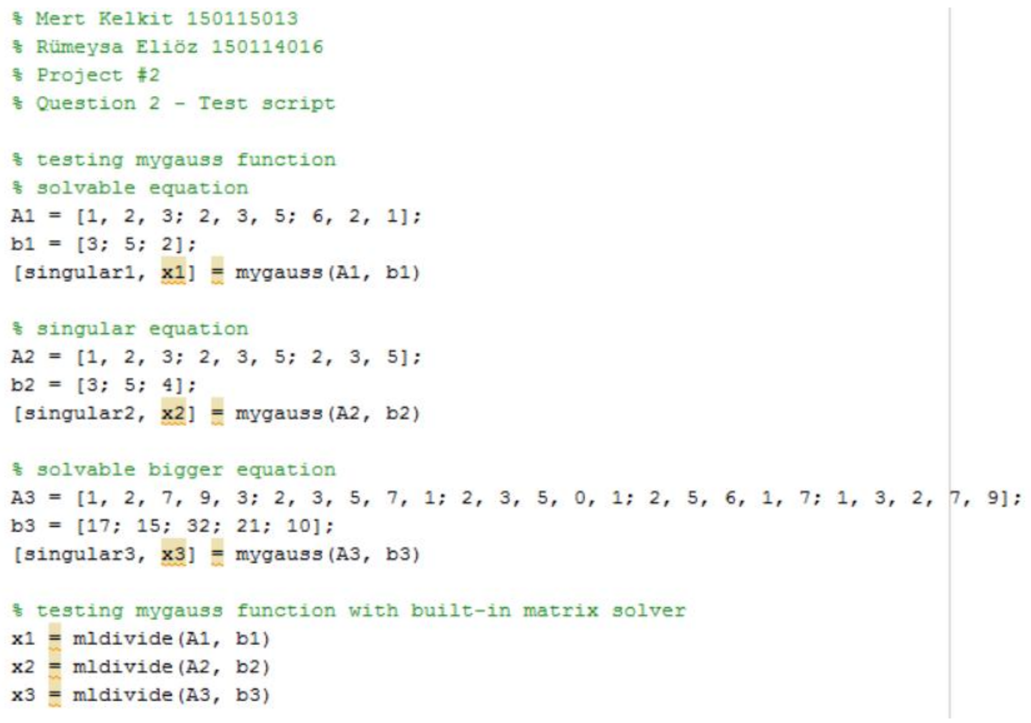
Test script for question 2:

Sample inputs to the function are : A1, b1

A2, b2 and

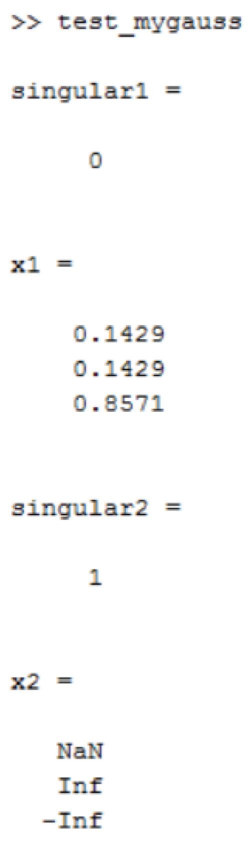
A3, b3.

Also we checked our results with the MATLAB built-in function “mldivide”.

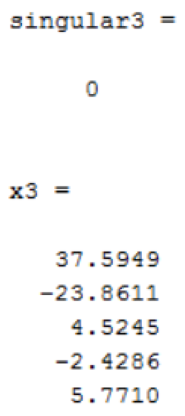


```
% Mert Kelkit 150115013
% Rümeysa Eliöz 150114016
% Project #2
% Question 2 - Test script

% testing mygauss function
% solvable equation
A1 = [1, 2, 3; 2, 3, 5; 6, 2, 1];
b1 = [3; 5; 2];
[singular1, x1] = mygauss(A1, b1)

% singular equation
A2 = [1, 2, 3; 2, 3, 5; 2, 3, 5];
b2 = [3; 5; 4];
[singular2, x2] = mygauss(A2, b2)

% solvable bigger equation
A3 = [1, 2, 7, 9, 3; 2, 3, 5, 7, 1; 2, 3, 5, 0, 1; 2, 5, 6, 1, 7; 1, 3, 2, 7, 9];
b3 = [17; 15; 32; 21; 10];
[singular3, x3] = mygauss(A3, b3)

% testing mygauss function with built-in matrix solver
x1 = mldivide(A1, b1)
x2 = mldivide(A2, b2)
x3 = mldivide(A3, b3)
```

Sample outputs from function [singular, x]mygauss(A, b):

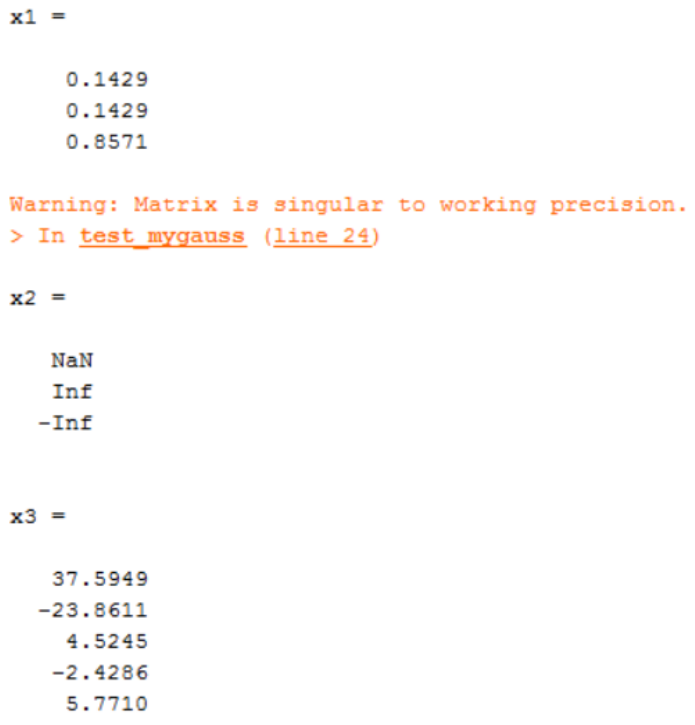


```
>> test_mygauss

singular1 =

     0


x1 =

    0.1429
    0.1429
    0.8571


singular2 =

     1


x2 =

   NaN
   Inf
  -Inf
```



```
singular3 =

     0


x3 =

   37.5949
  -23.8611
    4.5245
   -2.4286
    5.7710
```

Outputs from built-in function “mldivide”:



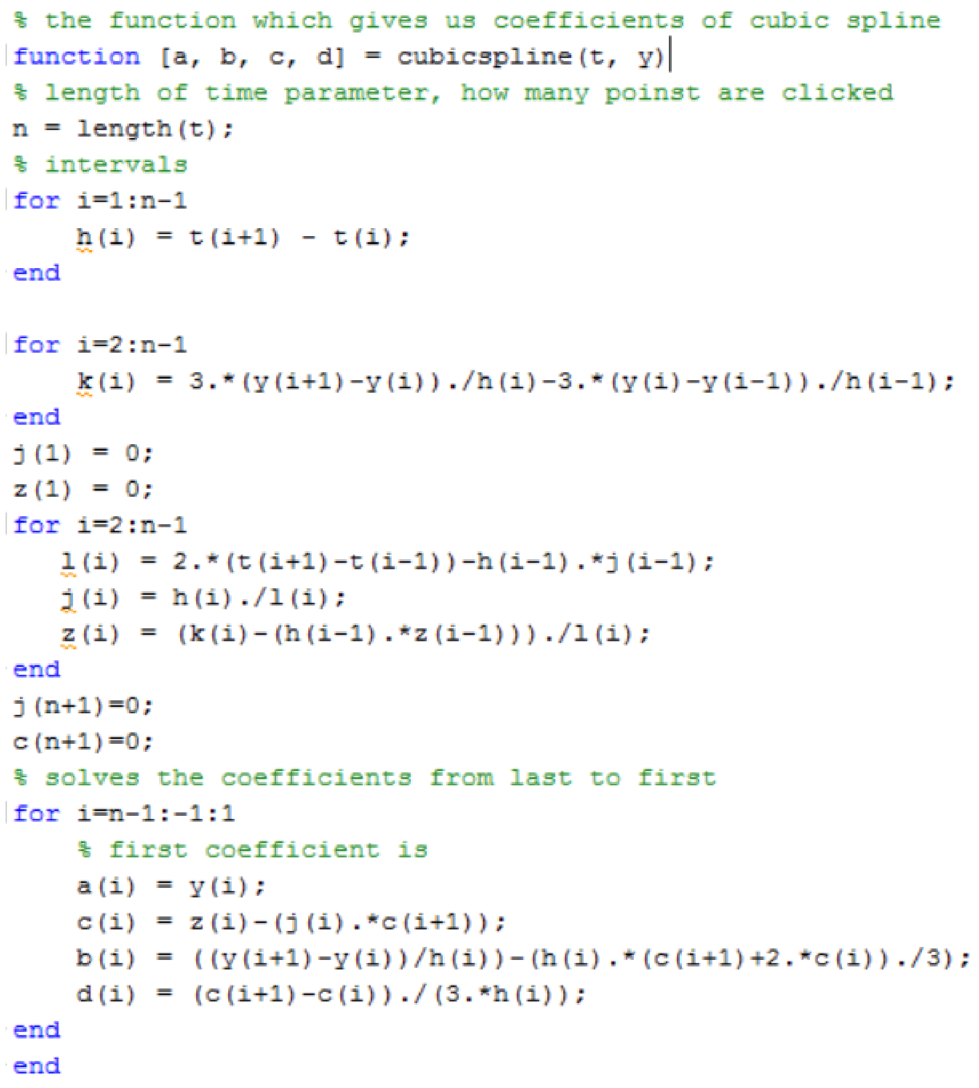
```
x1 =

    0.1429
    0.1429
    0.8571

Warning: Matrix is singular to working precision.
> In test_mygauss (line 24)

x2 =

   NaN
   Inf
  -Inf


x3 =

   37.5949
  -23.8611
    4.5245
   -2.4286
    5.7710
```

We saw that our function works fine. Warning says that we give a singular linear equation as an input which is also true for our case.

3) In this question, we have to create a function cubicspline which creates intervals with points x or y depending on time, and calculates the cubic spline equation coefficients a, b, c and d and returns them.

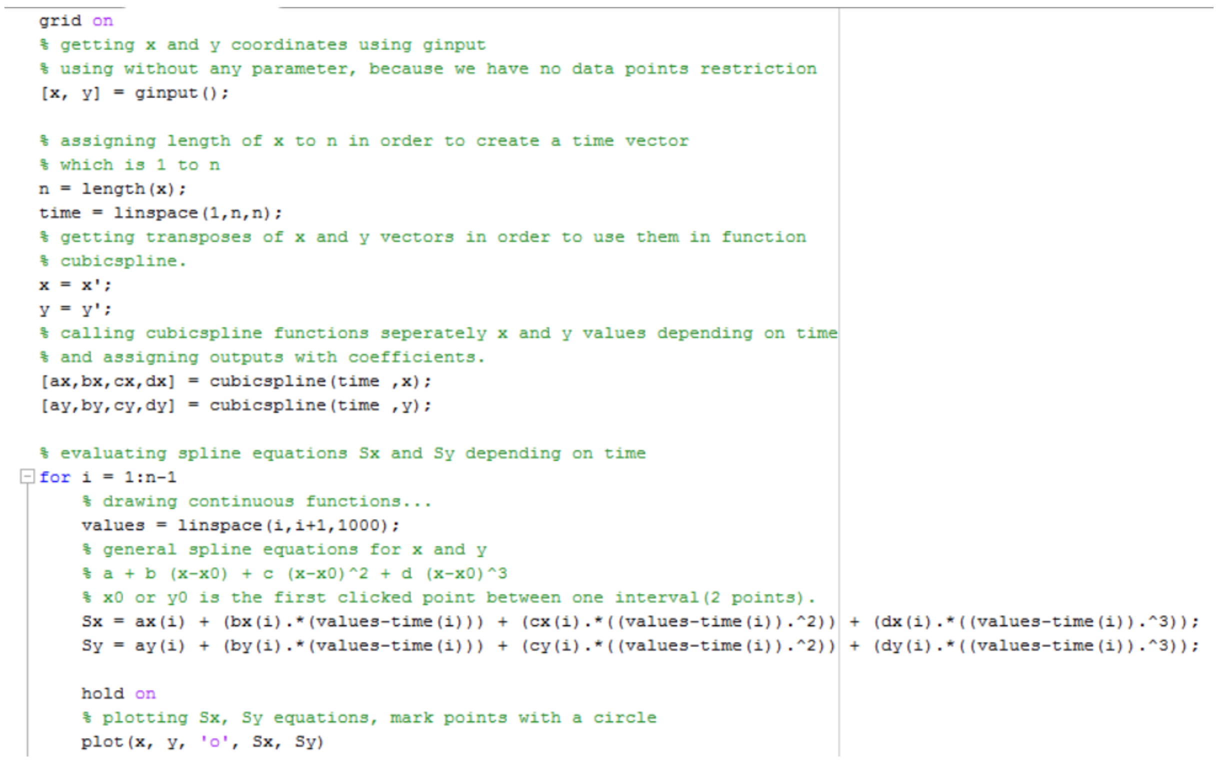
In this function, we get a time vector and also x or y coordinate vector. Then we create intervals with these two vectors. After creating spline equations

(e.g. $S_0 = a + b\,(x-x_0) + c\,(x-x_0)^2 + d\,(x-x_0)^3$) ($x_0$ is the first clicked point for example, and we find first clicked x or y coordinates from the time vector), we can see that every spline has 4 unknowns.(For example, 3 splines requires 12 linear equations to solve the coefficients).



```matlab
% the function which gives us coefficients of cubic spline
function [a, b, c, d] = cubicspline(t, y)
% length of time parameter, how many poinst are clicked
n = length(t);
% intervals
for i=1:n-1
    h(i) = t(i+1) - t(i);
end

for i=2:n-1
    k(i) = 3.*(y(i+1)-y(i))./h(i)-3.*(y(i)-y(i-1))./h(i-1);
end
j(1) = 0;
z(1) = 0;
for i=2:n-1
   l(i) = 2.*(t(i+1)-t(i-1))-h(i-1).*j(i-1);
   j(i) = h(i)./l(i);
   z(i) = (k(i)-(h(i-1).*z(i-1)))./l(i);
end
j(n+1)=0;
c(n+1)=0;
% solves the coefficients from last to first
for i=n-1:-1:1
    % first coefficient is
    a(i) = y(i);
    c(i) = z(i)-(j(i).*c(i+1));
    b(i) = ((y(i+1)-y(i))/h(i))-(h(i).*(c(i+1)+2.*c(i))./3);
    d(i) = (c(i+1)-c(i))./(3.*h(i));
end
end
```

After getting the coefficients from function cubicspline, we have to plot cubic splines in a script named “curvedraw”.



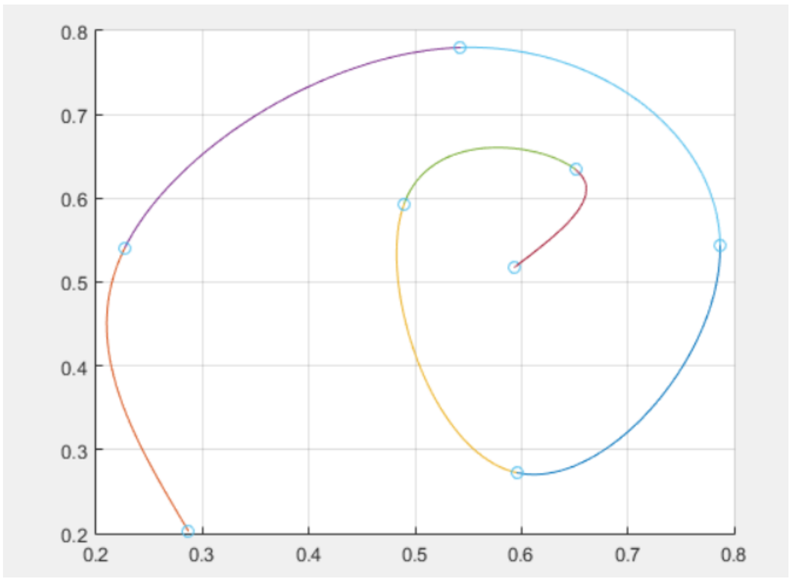
```
grid on
% getting x and y coordinates using ginput
% using without any parameter, because we have no data points restriction
[x, y] = ginput();

% assigning length of x to n in order to create a time vector
% which is 1 to n
n = length(x);
time = linspace(1,n,n);
% getting transposes of x and y vectors in order to use them in function
% cubicspline.
x = x';
y = y';
% calling cubicspline functions seperately x and y values depending on time
% and assigning outputs with coefficients.
[ax,bx,cx,dx] = cubicspline(time ,x);
[ay,by,cy,dy] = cubicspline(time ,y);

% evaluating spline equations Sx and Sy depending on time
for i = 1:n-1
    % drawing continuous functions...
    values = linspace(i,i+1,1000);
    % general spline equations for x and y
    % a + b (x-x0) + c (x-x0)^2 + d (x-x0)^3
    % x0 or y0 is the first clicked point between one interval(2 points).
    Sx = ax(i) + (bx(i).*(values-time(i))) + (cx(i).*((values-time(i)).^2)) + (dx(i).*((values-time(i)).^3));
    Sy = ay(i) + (by(i).*(values-time(i))) + (cy(i).*((values-time(i)).^2)) + (dy(i).*((values-time(i)).^3));

    hold on
    % plotting Sx, Sy equations, mark points with a circle
    plot(x, y, 'o', Sx, Sy)
```

We used ginput function without parameter in order to click and determine infinite number of points. After getting the clicked points by function ginput, we create a time vector which shows us which point is cliked first, helps us to create intervals etc. After calling the function cubicspline for x coordinates and y coordinates, we will get both coefficients ax, bx, cx, dx, ay, by, cy and dy. After that, we put these coefficients to the general spline equation in a for loop and calculate every spline equation as Sx and Sy. Then, mark clicked x and y points with a circle, plot splines Sx and Sy.

Sample input coordinates to function cubicspline and output plots from function cubicspline:

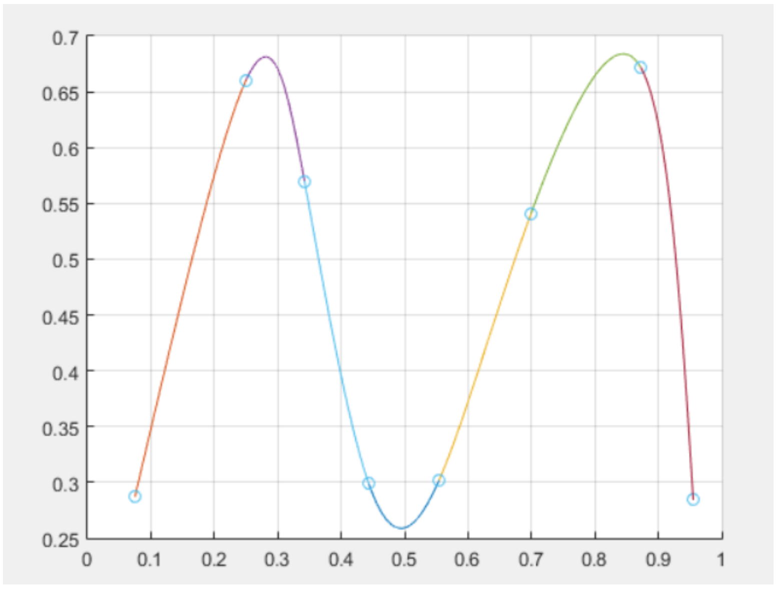
1-

```
x =

    0.2869
    0.2270
    0.5426
    0.7869
    0.5956
    0.4896
    0.6509
    0.5933


y =

    0.2026
    0.5408
    0.7799
    0.5437
    0.2726
    0.5933
    0.6341
    0.5175
```



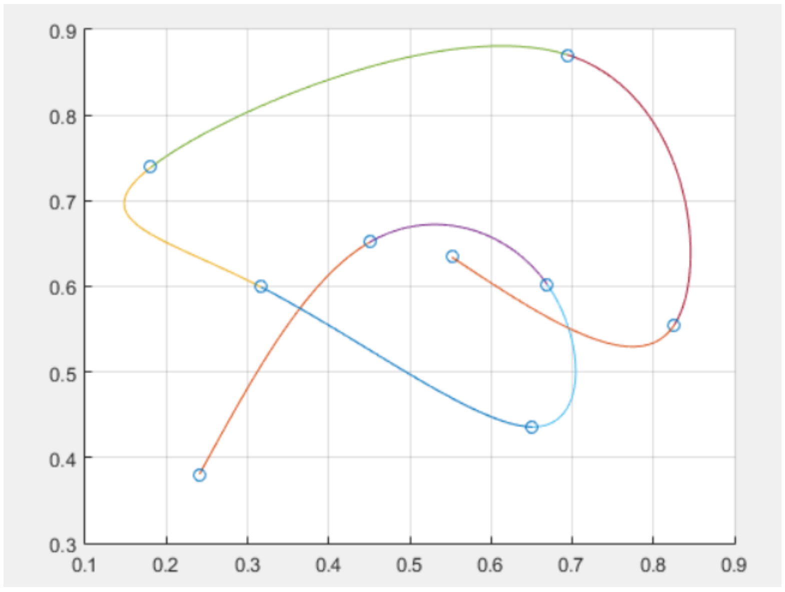
2-

```
x =

   0.0749
   0.2500
   0.3422
   0.4435
   0.5541
   0.6993
   0.8721
   0.9551


y =

   0.2872
   0.6603
   0.5700
   0.2988
   0.3017
   0.5408
   0.6720
   0.2843
```



3-

```
x =

    0.2408
    0.4505
    0.6694
    0.6509
    0.3168
    0.1809
    0.6947
    0.8260
    0.5518


y =

    0.3805
    0.6516
    0.6020
    0.4359
    0.5991
    0.7391
    0.8703
    0.5554
    0.6341
```



Sources

1- https://www.mathworks.com/

2- http://www.math.ucsd.edu/

3- R. L. Burden and J. D. Faires, Numerical Analysis, Brooks/Cole, 2011.

4- https://stackoverflow.com/

Mert KELKİT 150115013

Rümeysa ELİÖZ 150114016