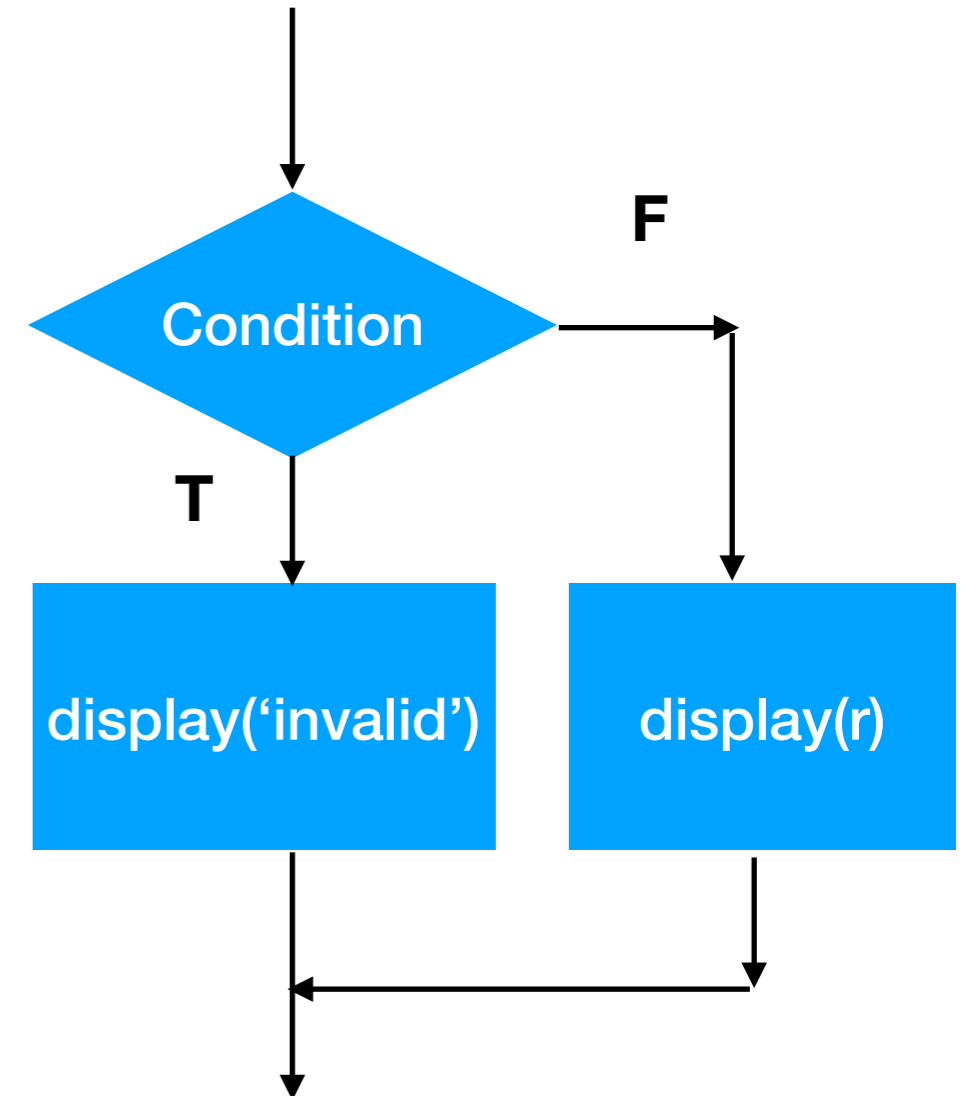


# Matlab control

第三行計算變數a與b的比值，當b為0時，該比值未定義，以Inf表示。請在條件指令中，將計算條件設定為判斷r是否等於Inf，在不同條件下分別執行不同命令。

```
1- a = 2;  
2- b = 0;  
3- r = a / b;  
4- if   
5-     display('invalid division')  
6- else  
7-     display(r)  
8- end
```



第三行計算變數a與b的比值，當比值r很接近0時，以r的絕對值是否小於eps來判斷r是否趨近於0，請依此原則設定if指令條件，在不同條件下分別執行不同命令。

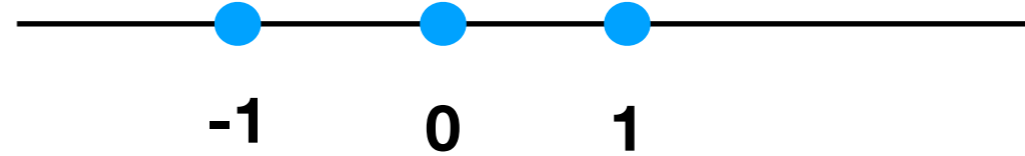
```
1 - a = -10^-6;  
2 - b = 10^10;  
3 - r = a / b;  
4 - if [REDACTED]  
5 -     display('near zero')  
6 - else  
7 -     display(r)  
8 - end
```

$$f(x) = \begin{cases} -x & \text{if } x < 0 \\ x & \text{if } 0 \leq x \end{cases}$$

當輸入值小於0時，函數f將輸出值設定為-x，否則設定為x

```
1 — x = 2;  
2 — if   
3 —     y = -x;  
4 —   
5 —     y = x;  
6 — end  
7 — display(y)
```

$$f(x) = \begin{cases} -1 & \text{if } x < -1 \\ -x & \text{if } -1 \leq x < 0 \\ x & \text{if } 0 \leq x < 1 \\ 1 & \text{if } 1 \leq x \end{cases}$$

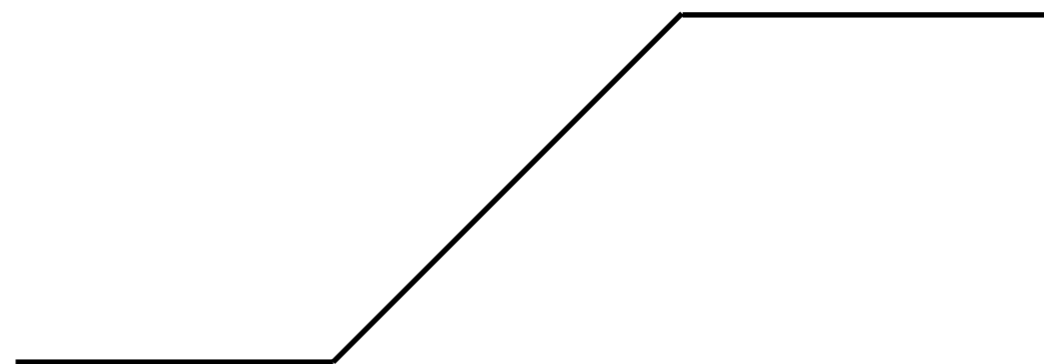


函數f的定義域有四個區間，根據輸入值x在不同區間，進行不同運算，f稱為piecewise函數，請在四行使用elseif完成函數f的計算

```

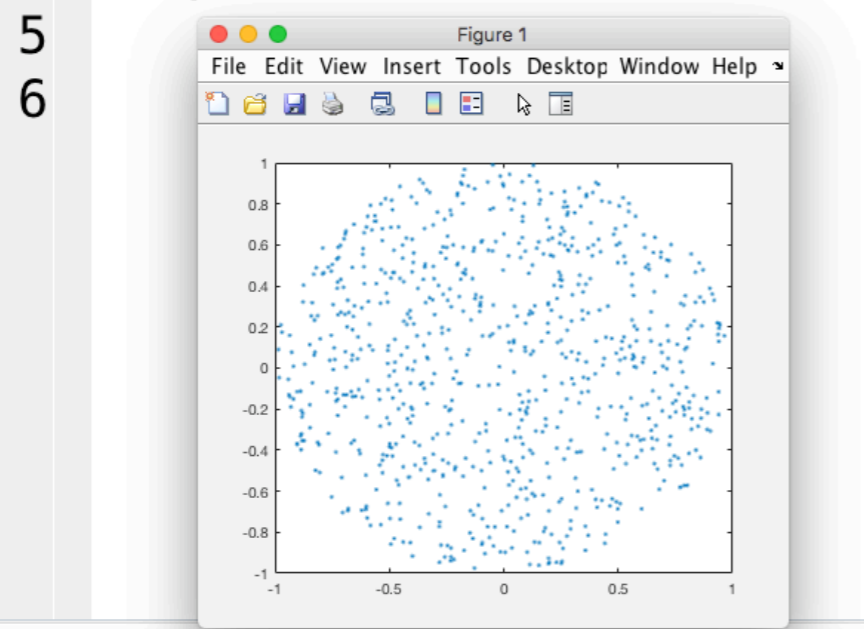
1 - x = 2;
2 - if x < -1
3 -     y = -1;
4 -
5 -     y = -x;
6 - elseif x < 1
7 -
8 - else
9 -     y = 1;
10 - end
11 - display(y)

```



**x**包含1000個分佈在 $[-1, 1] \times [-1, 1]$ 的二維點座標，請計算**r**向量，使向量元素代表二維點與原點的距離平方，使用**find**函數找出向量**r**中小於1的元素位置，本題繪製單位圓內的二維點

```
1 x = rand(2,1000)*2 - 1;  
2 r =           ;  
3 ind =           ;  
4 plot(x(1, ind), x(2, ind), 'b.')
```



**nums**儲存1到100間整數的隨機排列，請使用**find**指令找出整數**n**的位置

```
1 — nums = randperm(100);  
2 — n = 10;  
3 — ind =           ;  
4 — display(nums(ind))
```

# Coding Exercise

`a = 2; b = -1; n = 100`

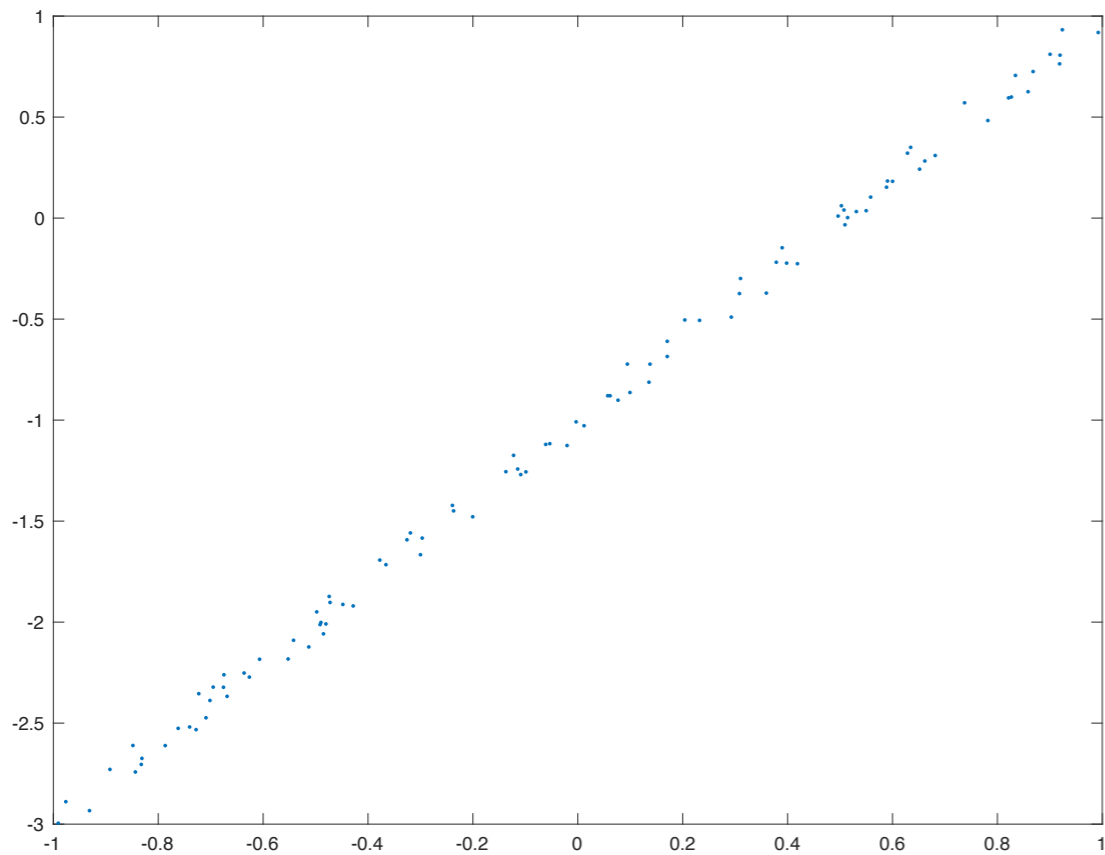
`x = 2*rand(n,1)-1; y = a * x + b + rand(n,1)*0.2-0.1;`

Use only one statement to estimate a and b

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix}$$



```
>> a = 2; b = -1; n = 100;  
>> x = 2*rand(n,1)-1;y = a * x + b+ rand(n,1)*0.2-0.1;  
>> plot(x,y, 'r')
```



$$y = 2x - 1$$

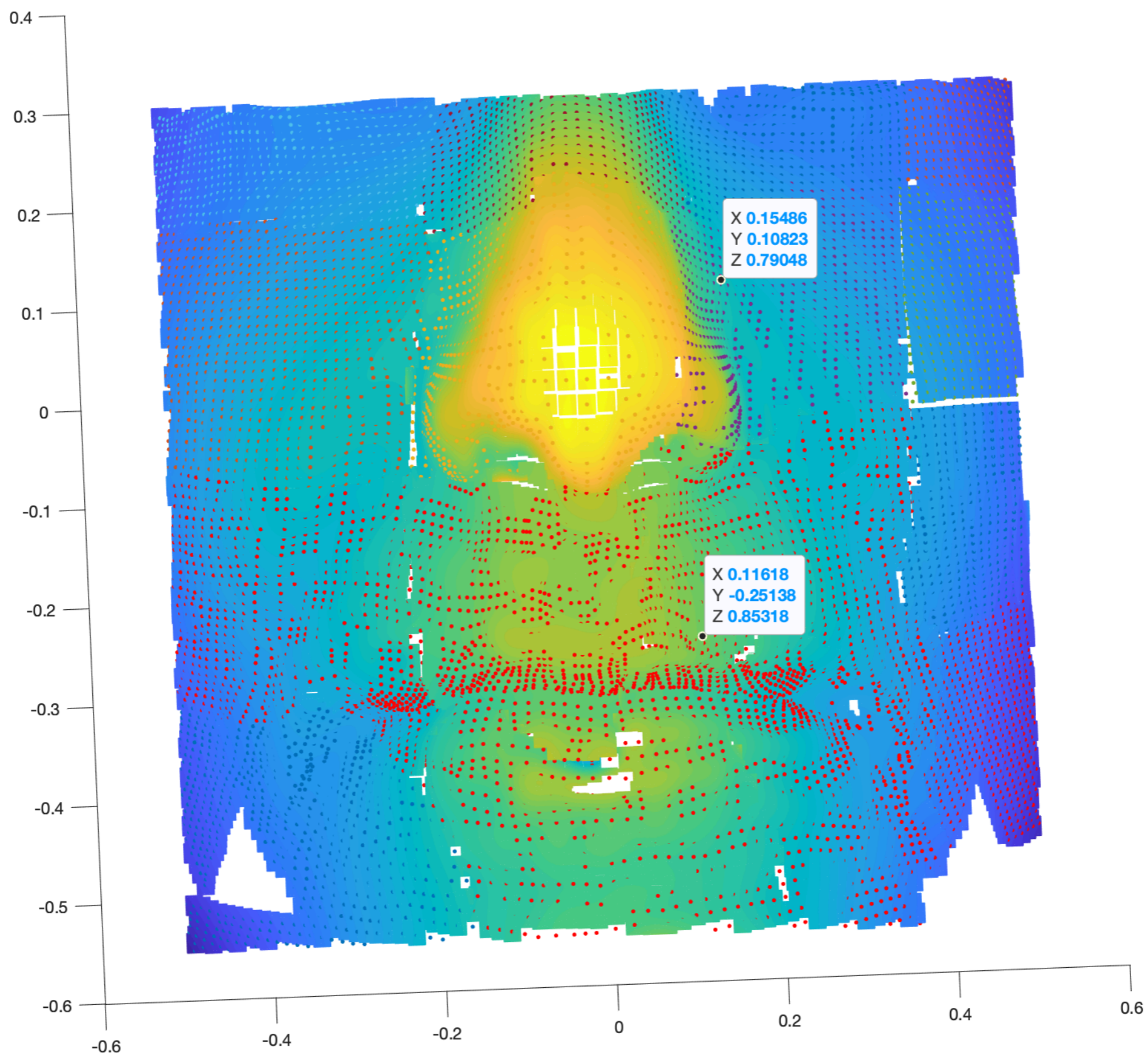
$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix}$$

sum(x.^2)
sum(x.\*y)

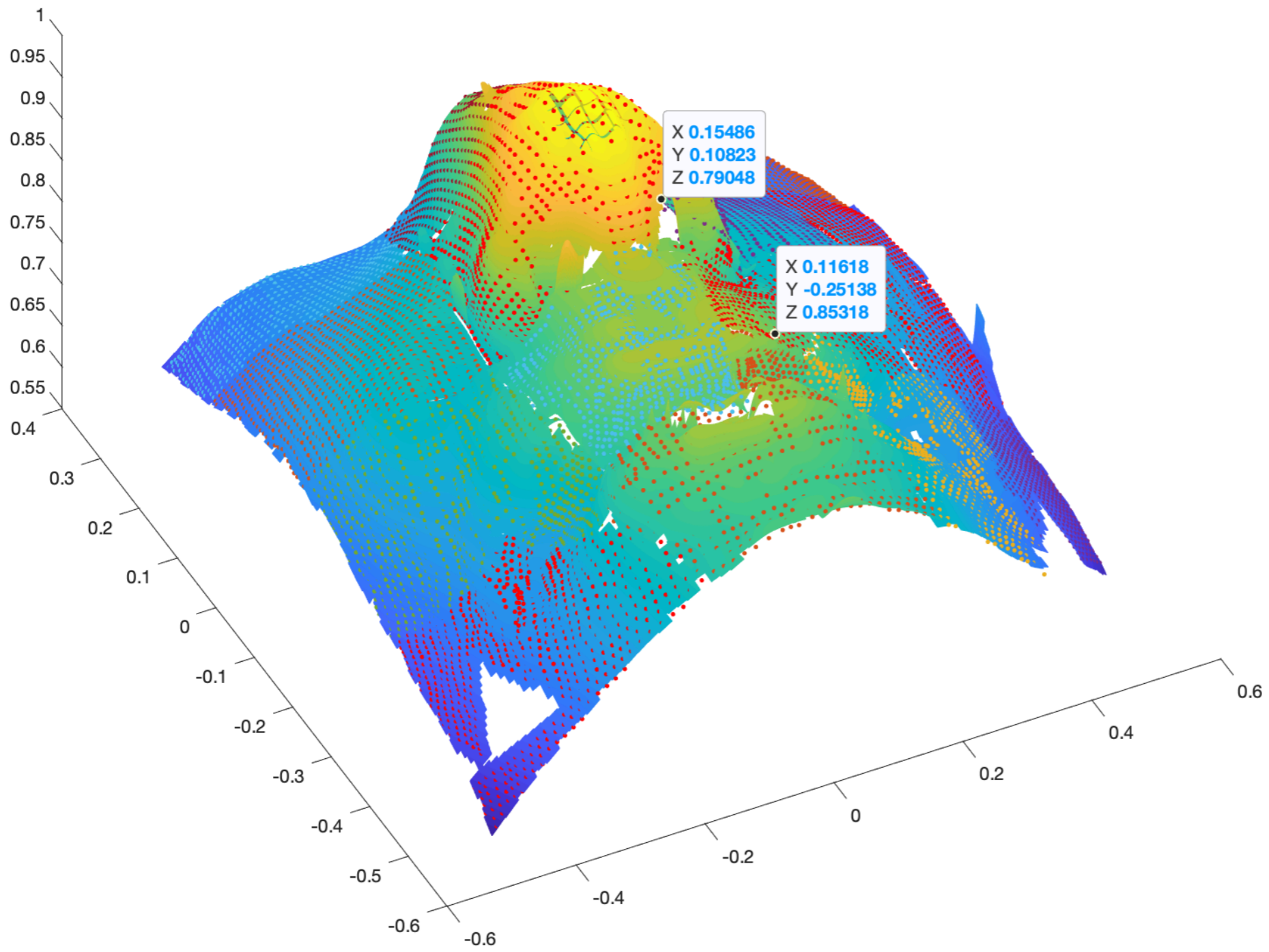
`inv([sum(x.^2) sum(x);sum(x) n])*transpose([sum(x.*y) sum(y)])`

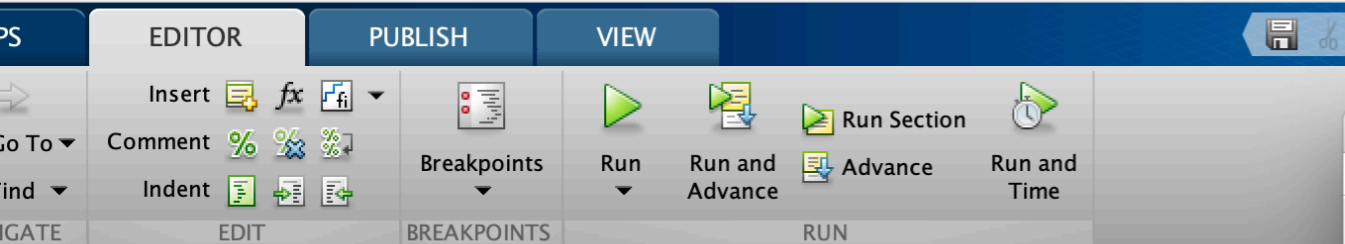
`inv([sum(x.^2) sum(x);sum(x) n])*[sum(x.*y) sum(y)]'`

# deep learning



# deep learning





apple ▶ Desktop ▶ Jiann-Ming Wu ▶ code2019 ▶ code2006 ▶ Apps ▶ NEN ▶ Large\_Scaled\_NEN

Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/code2006/Apps/NEN/L

demo\_plot\_subface\_deep.m x plot\_subface.m x demo\_faceNet\_p4\_plot.m

```
1 function plot_subface(faceNet,X,y)
2     Net = faceNet.Net;
3     Y_max = faceNet.Y_max;
4     best_beta = faceNet.best_beta;
5     x=X;
6     x_max = 1;
7     y= y';
8     tex=x;tey=y;
9     plot3(tex(:,1), tex(:,2), tey(:,1))
```

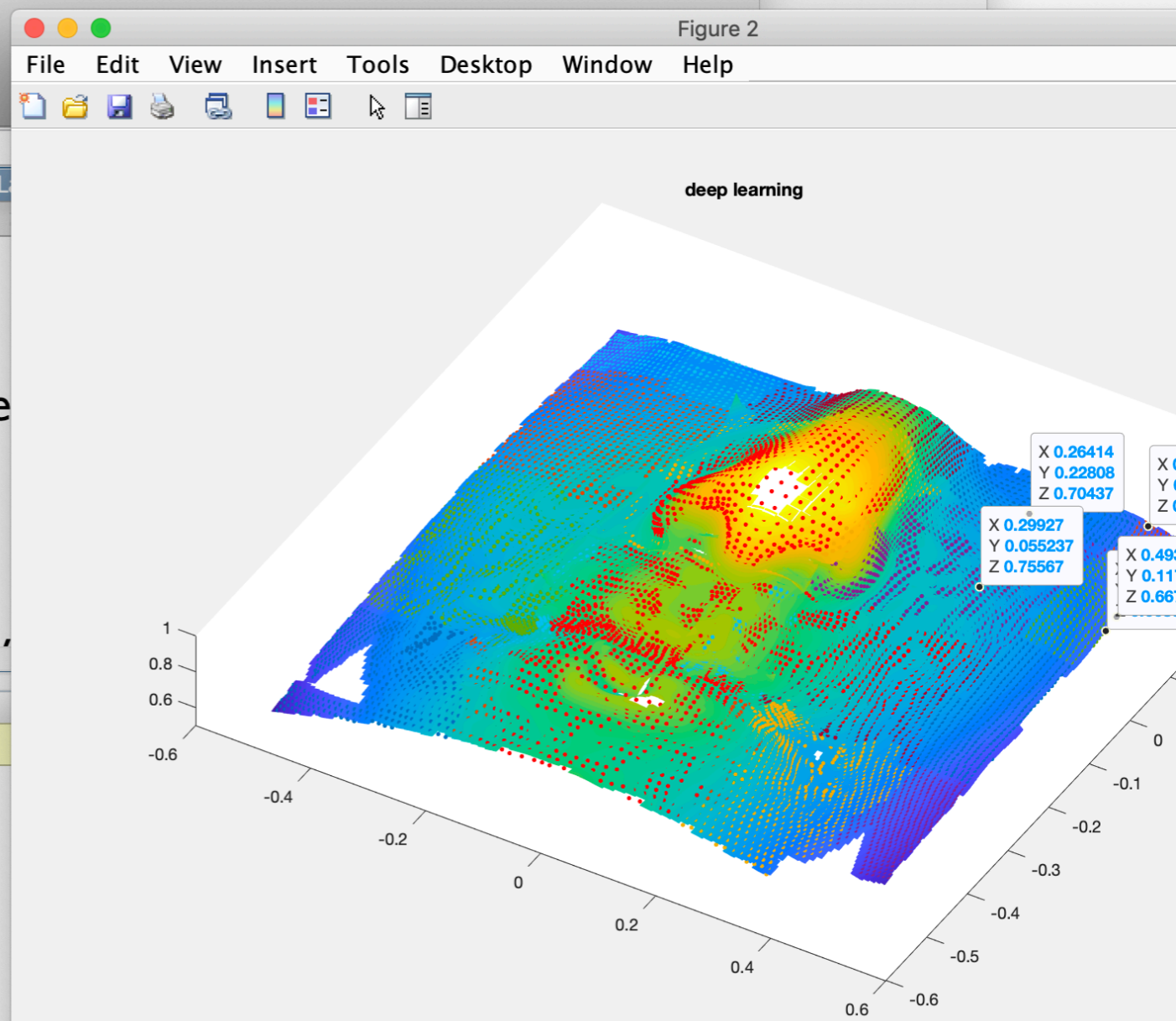
Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> openfig('faceNet_p4.fig')
```

```
ans =
```

[Figure](#) (2) with properties:



iphone

LetterCore | Build LetterCore: **Succeeded** | Today at 7:42 AM

LetterCore > LetterCore > EMNIST\_letter.mlmodel

LetterCore

- LetterCore
  - AppDelegate.swift
  - ViewController.swift
  - MainView.swift
  - ShadowView.swift
  - UIView+Constraints.swift
  - Float+String.swift
  - Colors.swift
  - EMNIST\_letter.mlmodel**
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
  - Products

**Machine Learning Model**

Name EMNIST\_letter  
Type Neural Network  
Size 9 MB  
Author C.Tien  
Description This CNN model without BatchNorm is for testing of English handwritten-alphabet recognition.  
License unknown

**Model Class**

EMNIST\_letter  
Automatically generated Swift model class

**Prediction**

Name	Type	Description
▼ Inputs		
data	MultiArray (Double 1 x 784 x 1)	Input image multiarray
▼ Outputs		
prob	MultiArray (Double)	output A-Z element

**Identity and Type**

Name EMNIST\_letter.mlmodel  
Type Default - CoreML Model  
Location Relative to Group  
EMNIST\_letter.mlmodel  
Full Path /Users/apple/Desktop/swift\_2019/LetterCore36\_\_demo/LetterCore/EMNIST\_letter.mlmodel

**On Demand Resource Tags**

Only resources are taggable

**Localization**

Localize...

**Target Membership**

- LetterCore

mlmodel

Running on iPhone

The screenshot shows the Xcode IDE with a project named "LetterCore" selected. The project navigator on the left shows the file structure, with "EMNIST\_letter.mlmodel" selected. The central editor displays the details of this model:

- Machine Learning Model**
  - Name: EMNIST\_letter
  - Type: Neural Network
  - Size: 9 MB
  - Author: C.Tien
  - Description: This CNN model without BatchNorm is for testing of English handwritten-alphabet recognition.
  - License: unknown
- Model Class**
  - EMNIST\_letter (Automatically generated Swift model class)
- Prediction**

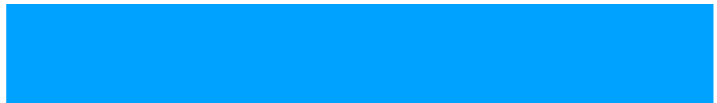
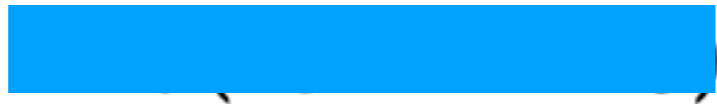
Name	Type	Description
▼ Inputs		
data	MultiArray (Double 1 x 784 x 1)	Input image multiarray
▼ Outputs		
prob	MultiArray (Double)	output A-Z element

The right-hand sidebar shows the "Identity and Type" section with the following details:

- Name: EMNIST\_letter.mlmodel
- Type: Default - CoreML Model
- Location: Relative to Group
- EMNIST\_letter.mlmodel
- Full Path: /Users/apple/Desktop/swift\_2019/LetterCore36\_demo/LetterCore/EMNIST\_letter.mlmodel

At the bottom of the interface, the status bar shows the date and time "2019-10-14 07:48:50.460634+0800" and the text "LetterCore[2215:8289201] Metal API Validation Enabled".

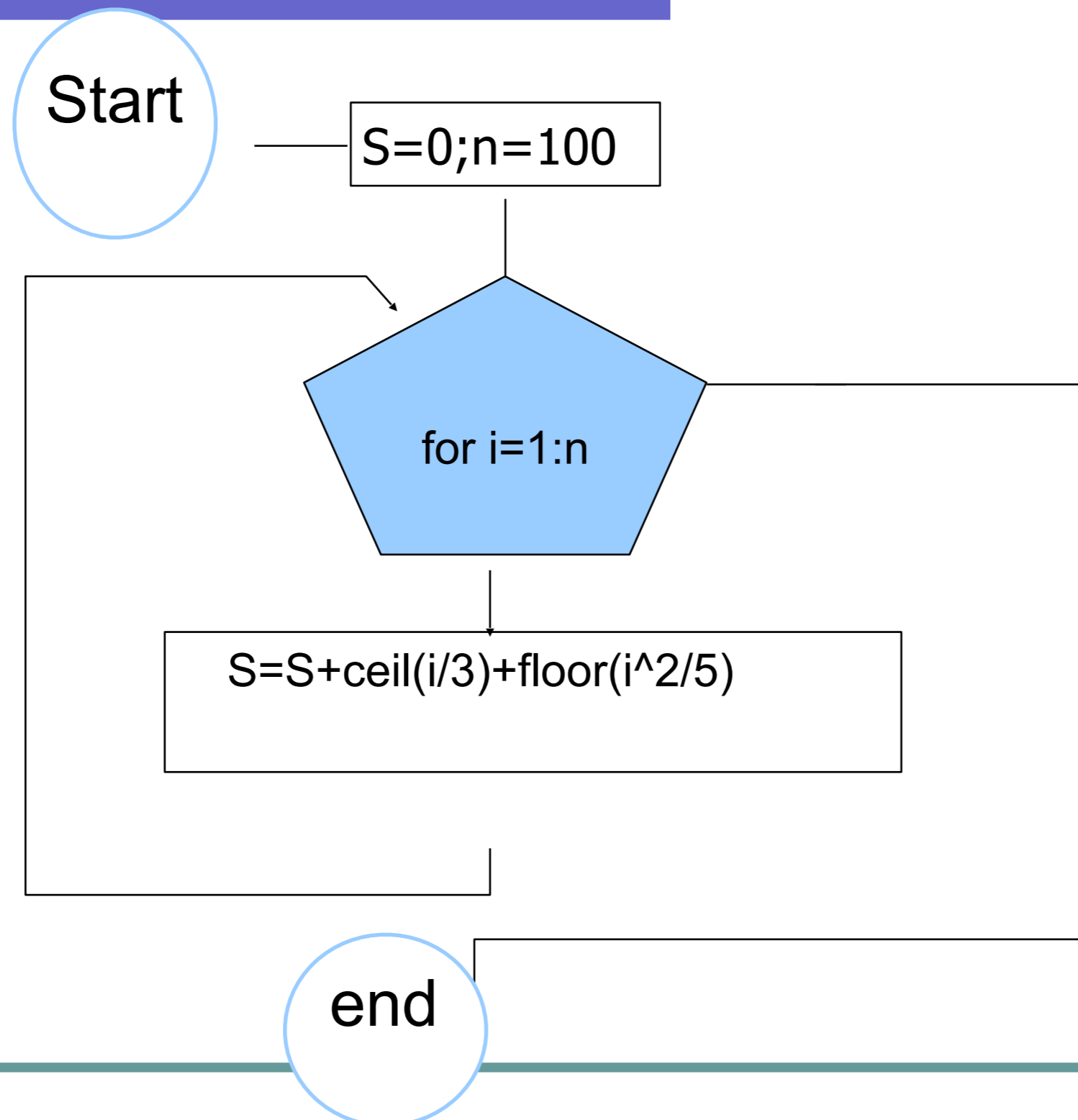
**nums**儲存1到100間整數的隨機排列，請使用指令**mod**，求**nums**所有元素除以17的餘數，**find**指令找出所有餘數為0的位置，本題印出17的倍數排列

```
nums = randperm(100);  
rem = ;  
ind = ;  
display(nums(ind))  
17 68 34 85 51
```



# Flow chart

$$S(N) = \sum_{n=1}^N \left( \left\lfloor \frac{n^2}{5} \right\rfloor + \left\lceil \frac{2 * n}{3} \right\rceil \right)$$



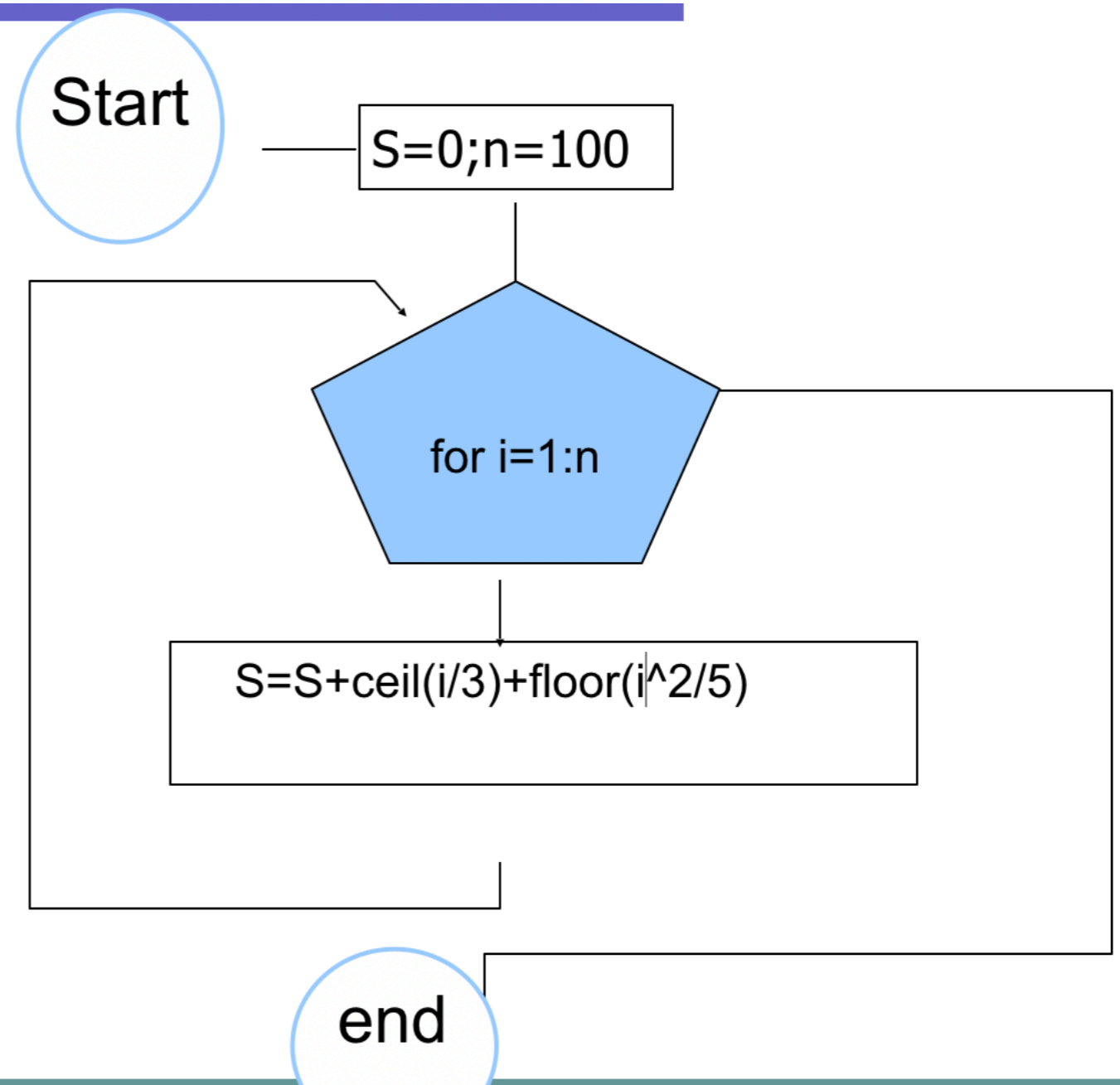
$$\sum_{i=1}^{100} \left( \left\lceil \frac{i}{3} \right\rceil + \left\lfloor \frac{i^2}{5} \right\rfloor \right)$$

請設定迴圈索引，並使用ceil與floor計算上式總和

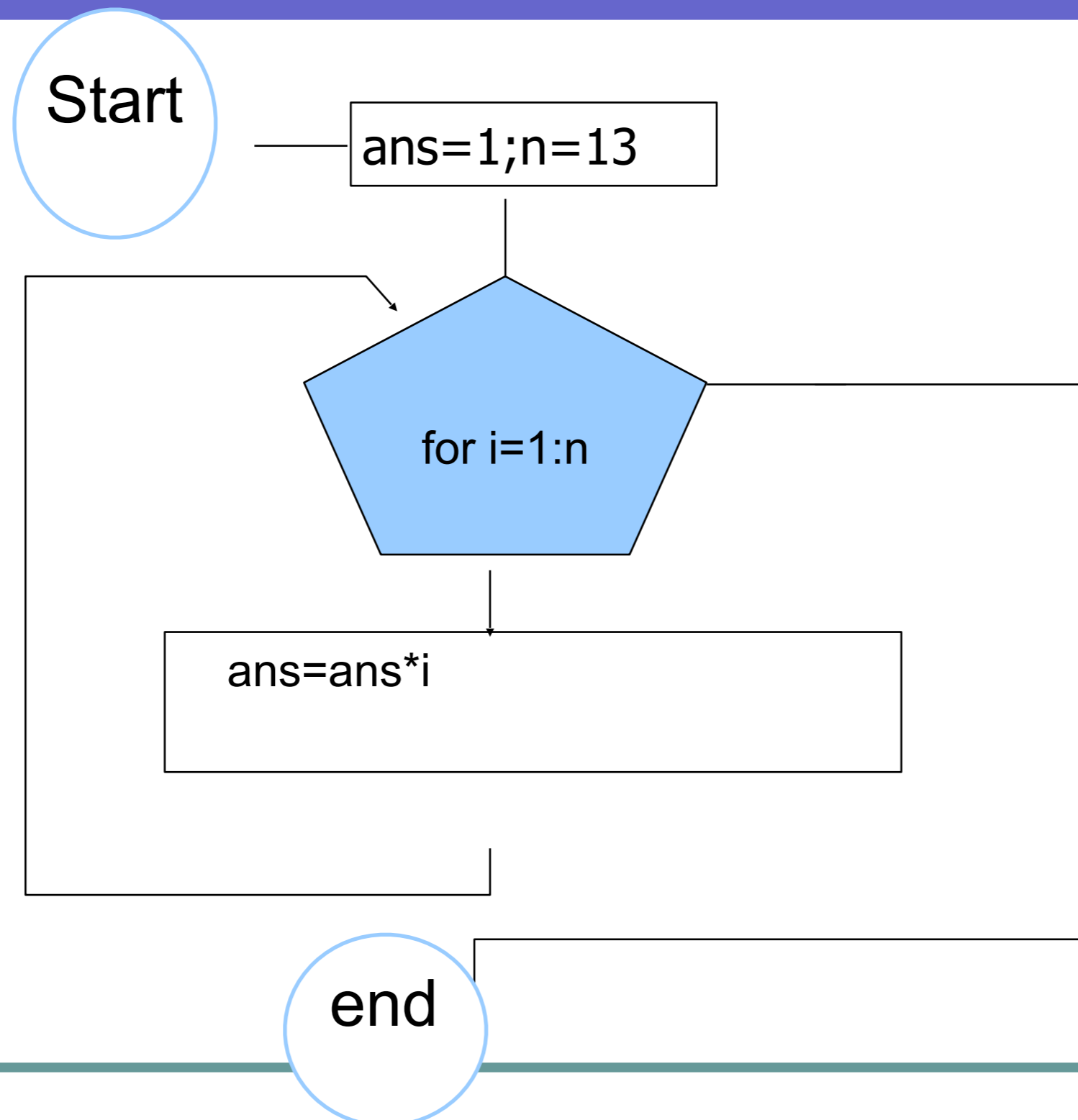
```
s = 0;  
n = 100;  
for i =   
    s =   
end  
display(s)
```

S =

69347



# Flow chart



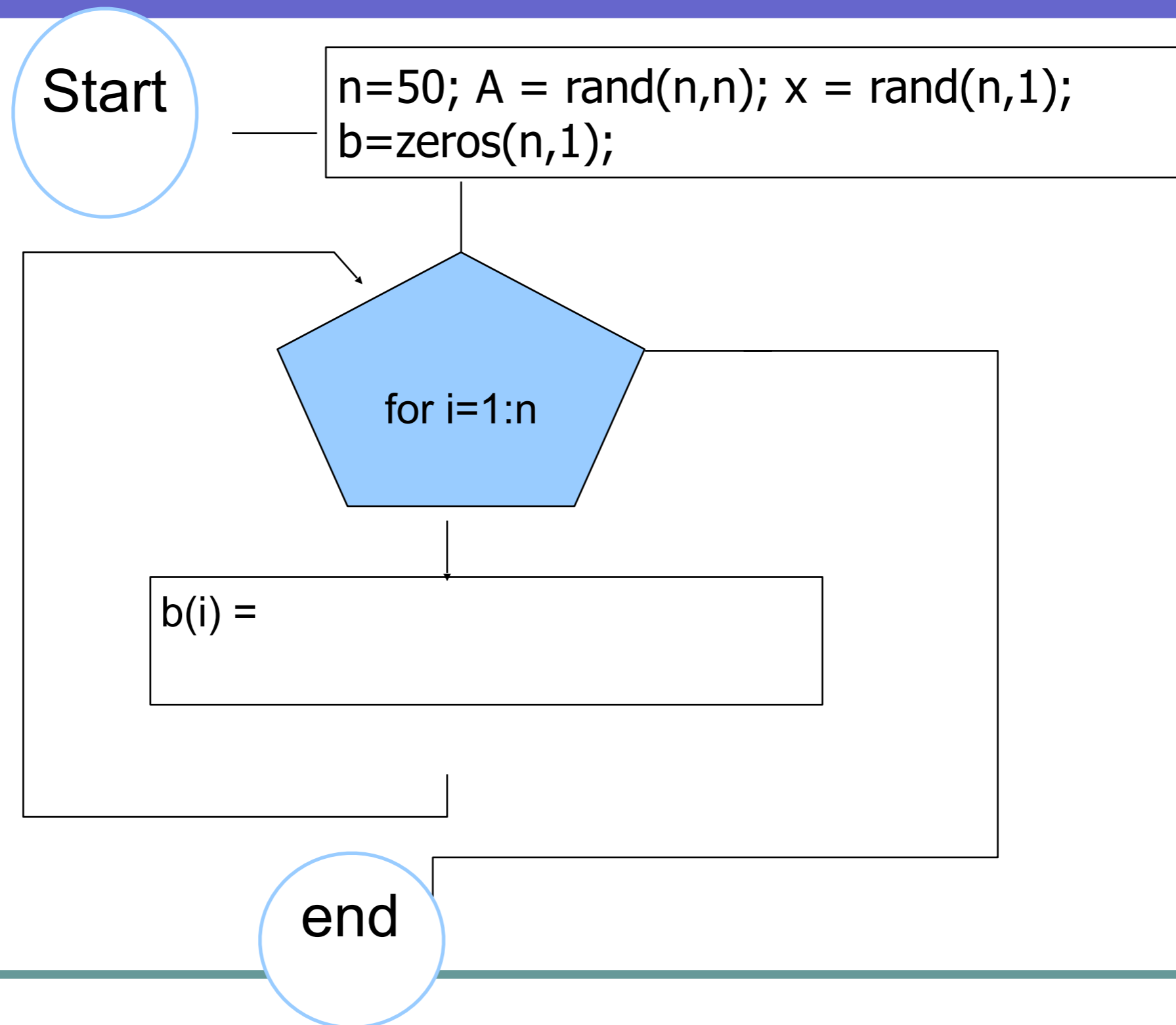
本題計算n!，請設定變數ans的初始值，並在迴圈內更新ans內容

```
    _____;  
n = 13;  
for i = 1 : n  
    ans = _____;  
end  
display(ans)
```

```
ans =
```

```
6.2270208000000000e+09
```

# Flow chart



本題計算矩陣A與直列向量x的乘績，請在迴圈主體命令中，計算A的第i橫列與直列向量x的乘績，在最後一行計算驗證A與x的乘績是否等於b

```
n = 50;  
A = rand(n, n);  
x = rand(n, 1);  
b = zeros(n, 1);  
for i = 1 : n  
    b(i) =                     ;  
end  
sum(abs(                    ))
```

ans =

8.348877145181177e-14

本題以A矩陣的第一橫列的A(1, 1)元素為基準，進行列運算，消去第二、三、四列的第一個元素，請在迴圈主體命令中計算比值，進行列運算

```
A = reshape(1 : 16, 4, 4);  
for i = 2 : 4  
    ratio = _____;  
    A(i, :) = _____;  
end  
display(A)
```

```
A =  
  
    1     5     9    13  
    0    -4    -8   -12  
    0    -8   -16   -24  
    0   -12   -24   -36
```

```
A =  
  
     1     5     9    13  
     2     6    10    14  
     3     7    11    15  
     4     8    12    16
```