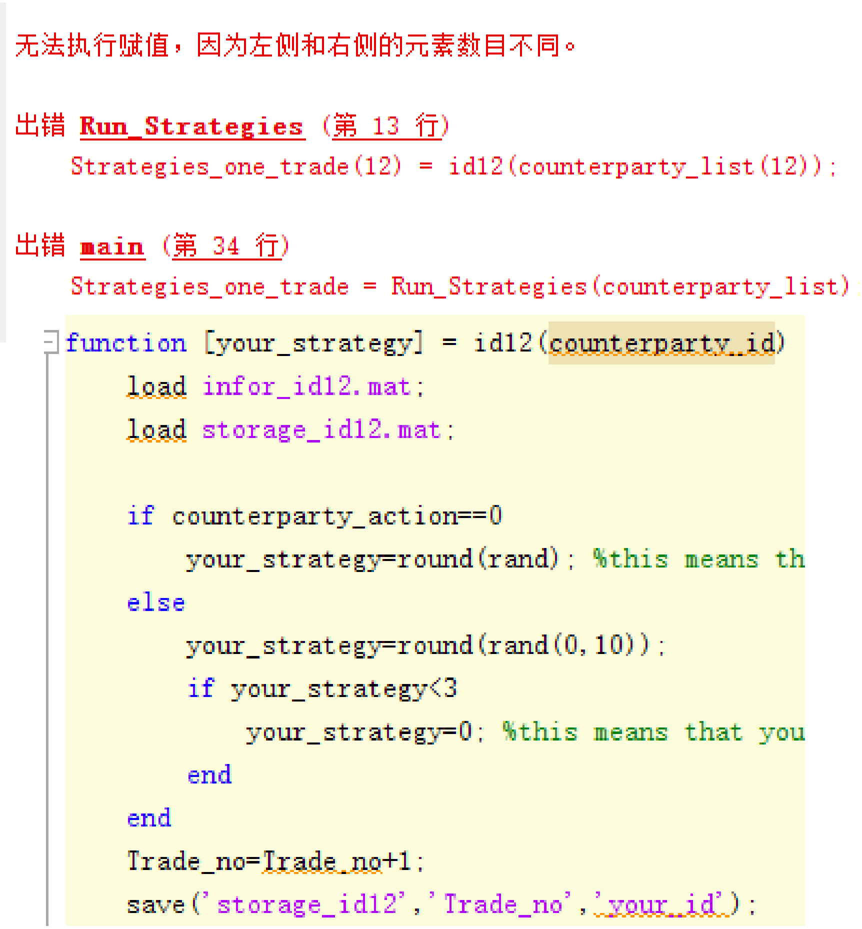
Stochastic Signal Processing

Lesson 2 – Experiment: changing a simulation system

Weize Sun

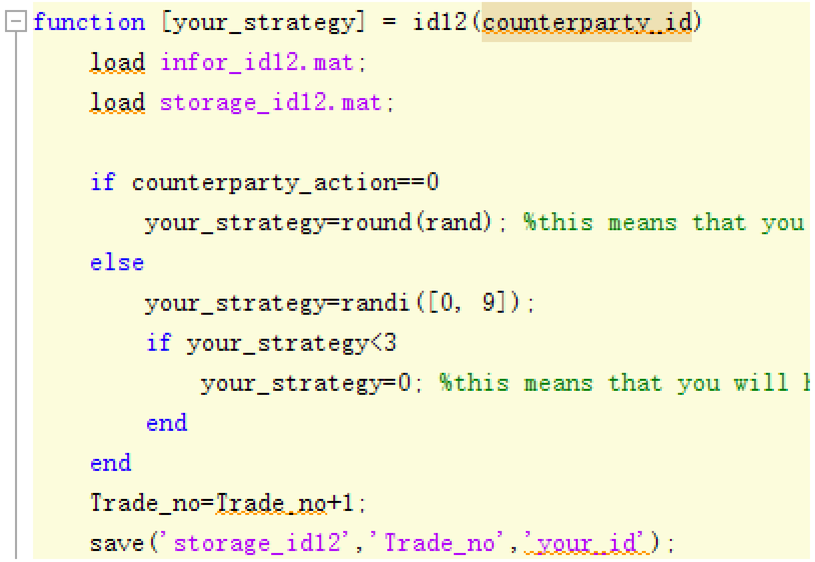
Problem of codes (this year)

- There are some problems of the submitted program:
    - data format error



```
无法执行赋值，因为左侧和右侧的元素数目不同。

出错 Run_Strategies (第 13 行)
    Strategies_one_trade(12) = id12(counterparty_list(12));

出错 main (第 34 行)
    Strategies_one_trade = Run_Strategies(counterparty_list);
```

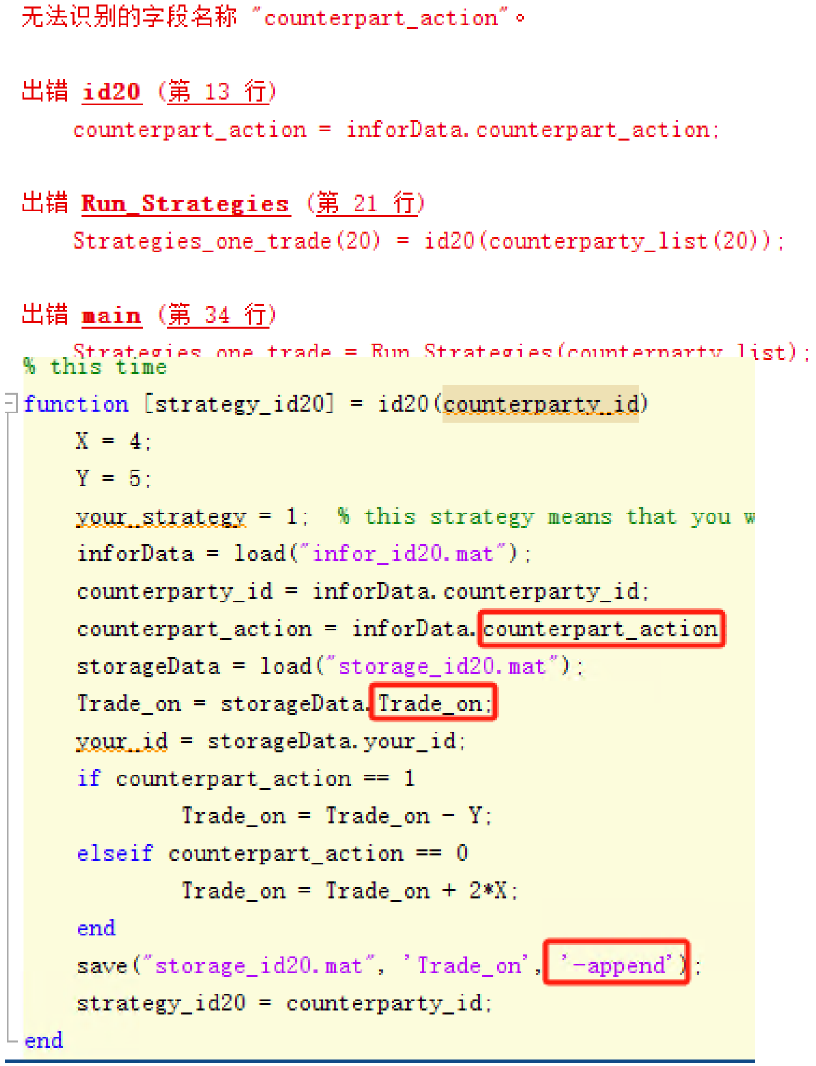
```
function [your_strategy] = id12(counterparty_id)
    load infor_id12.mat;
    load storage_id12.mat;

    if counterparty_action==0
        your_strategy=round(rand); %this means th
    else
        your_strategy=round(rand(0,10));
        if your_strategy<3
            your_strategy=0; %this means that you
        end
    end
    Trade_no=Trade_no+1;
    save('storage_id12','Trade_no','your_id');
```



```
function [your_strategy] = id12(counterparty_id)
    load infor_id12.mat;
    load storage_id12.mat;

    if counterparty_action==0
        your_strategy=round(rand); %this means that you
    else
        your_strategy=randi([0, 9]);
        if your_strategy<3
            your_strategy=0; %this means that you will l
        end
    end
    Trade_no=Trade_no+1;
    save('storage_id12','Trade_no','your_id');
```

Problem of codes (this year)

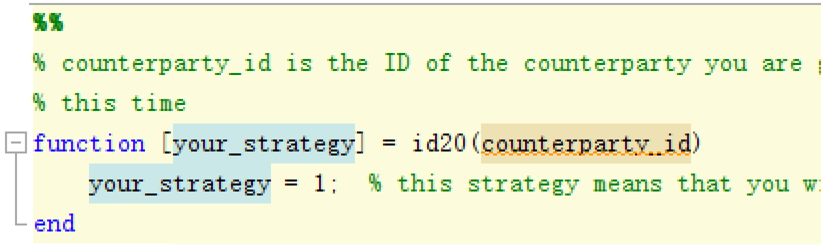
- There are some problems of the submitted program:
  - Wrong parameter name



```
无法识别的字段名称 "counterpart_action"。

出错 id20 (第 13 行)
    counterpart_action = inforData.counterpart_action;

出错 Run_Strategies (第 21 行)
    Strategies_one_trade(20) = id20(counterparty_list(20));

出错 main (第 34 行)
    Strategies_one_trade = Run_Strategies(counterparty_list);
```

```
% this time
function [strategy_id20] = id20(counterparty_id)
    X = 4;
    Y = 5;
    your_strategy = 1;  % this strategy means that you w
    inforData = load("infor_id20.mat");
    counterparty_id = inforData.counterparty_id;
    counterpart_action = inforData.counterpart_action
    storageData = load("storage_id20.mat");
    Trade_on = storageData.Trade_on;
    your_id = storageData.your_id;
    if counterpart_action == 1
            Trade_on = Trade_on - Y;
    elseif counterpart_action == 0
            Trade_on = Trade_on + 2*X;
    end
    save("storage_id20.mat", 'Trade_on', '-append');
    strategy_id20 = counterparty_id;
end
```

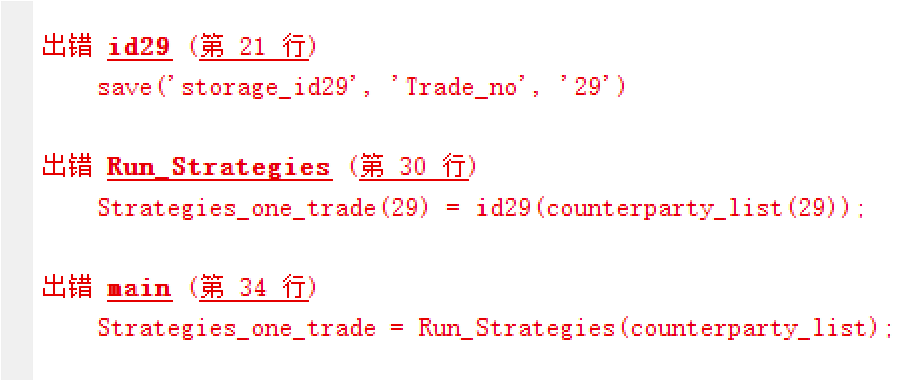
First is the error in the variable names:
counterparty_action
Trade_no
Second, the code is difficult to understand, recommended to provide comments, which makes the code easier to understand



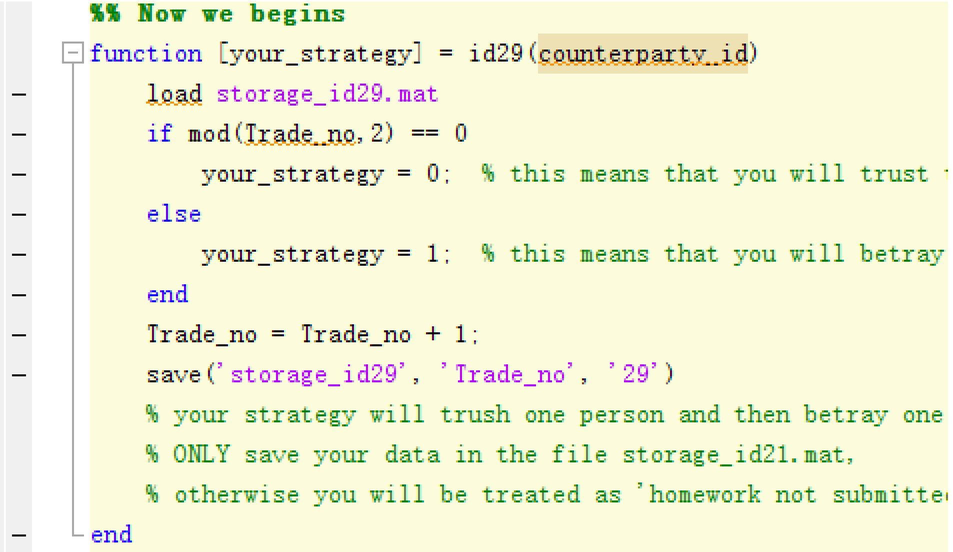
```
%%
% counterparty_id is the ID of the counterparty you are
% this time
function [your_strategy] = id20(counterparty_id)
    your_strategy = 1;  % this strategy means that you w
end
```

Problem of codes (this year)

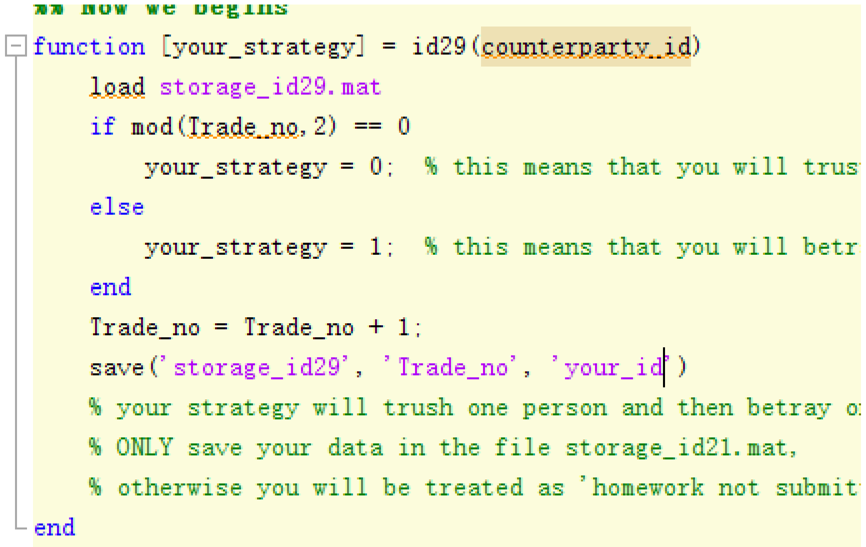
- There are some problems of the submitted program:
  - Saving strange things



```
出错 id29 (第 21 行)
    save('storage_id29', 'Trade_no', '29')

出错 Run_Strategies (第 30 行)
    Strategies_one_trade(29) = id29(counterparty_list(29));

出错 main (第 34 行)
    Strategies_one_trade = Run_Strategies(counterparty_list);
```



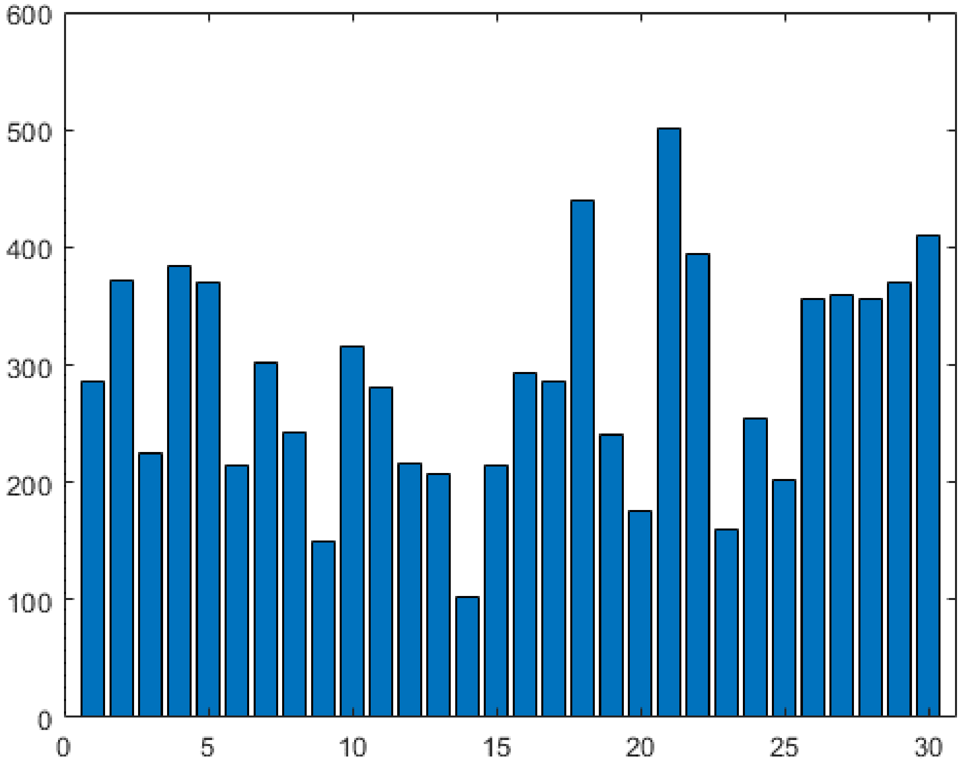
```
%% Now we begins
function [your_strategy] = id29(counterparty_id)
    load storage_id29.mat
    if mod(Trade_no,2) == 0
        your_strategy = 0;  % this means that you will trust t
    else
        your_strategy = 1;  % this means that you will betray
    end
    Trade_no = Trade_no + 1;
    save('storage_id29', 'Trade_no', '29')
    % your strategy will trush one person and then betray one
    % ONLY save your data in the file storage_id21.mat,
    % otherwise you will be treated as 'homework not submitte
end
```



```
%% Now we begins
function [your_strategy] = id29(counterparty_id)
    load storage_id29.mat
    if mod(Trade_no,2) == 0
        your_strategy = 0;  % this means that you will trus
    else
        your_strategy = 1;  % this means that you will betr
    end
    Trade_no = Trade_no + 1;
    save('storage_id29', 'Trade_no', 'your_id')
    % your strategy will trush one person and then betray o
    % ONLY save your data in the file storage_id21.mat,
    % otherwise you will be treated as 'homework not submit
end
```

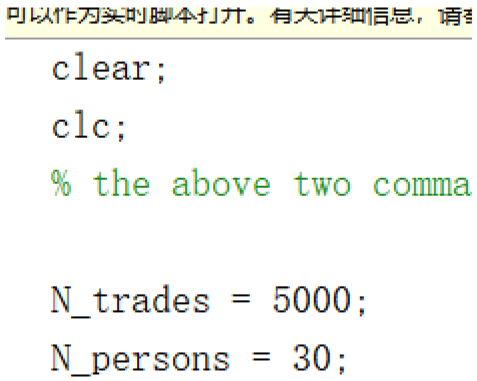
Results from Last Game

- Run 100 times among 30 persons
  - Note that the result will be different every time, as 100 trades is a small number



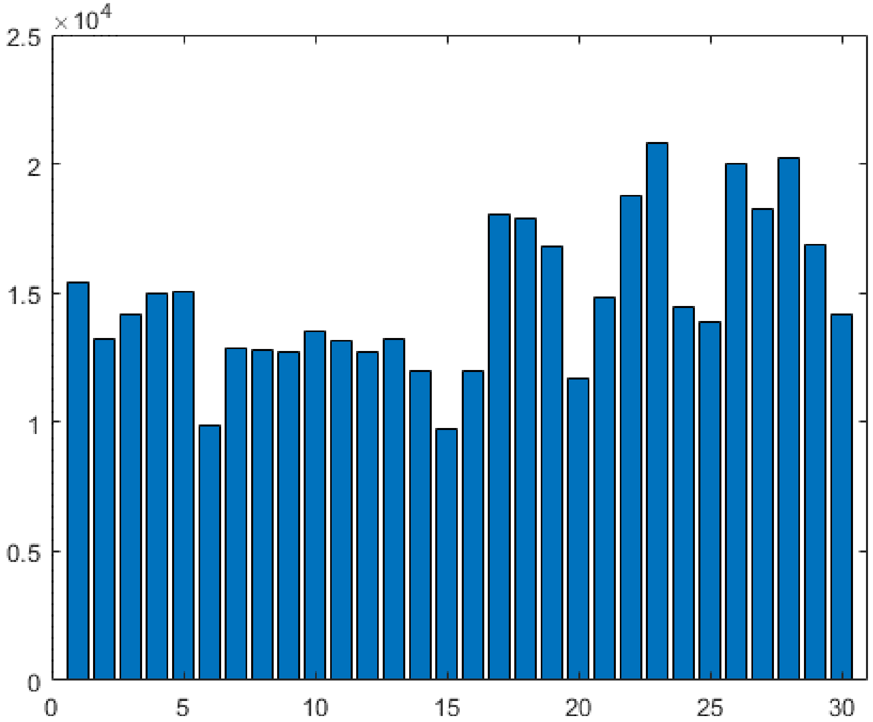
Results from Last Game

- Run 5000 times among 30 persons
  - If you run this again, you will see very similar result
  - This is the origin of the so called ‘Monte Carlo’ Simulation: test your model with an extremely long time (here, it means trade a large number of times), and give the average performance
  - However, a person cannot trade with others 5000 times in real life, that is unrealistic



可以作为实时脚本打开。有关详细信息，请

```
clear;
clc;
% the above two comma

N_trades = 5000;
N_persons = 30;
```

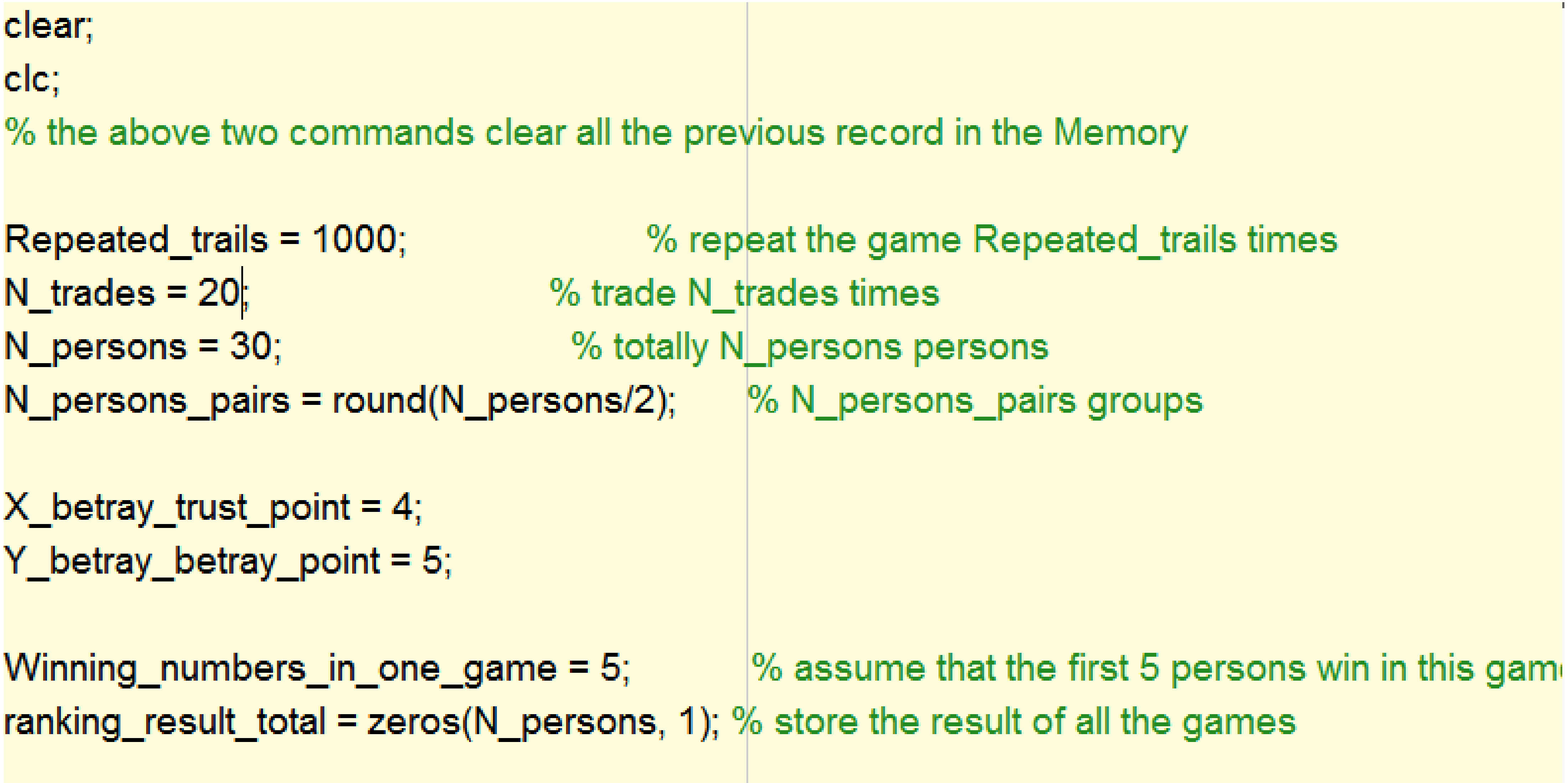


Results from Last Game

Then how to be realistic?

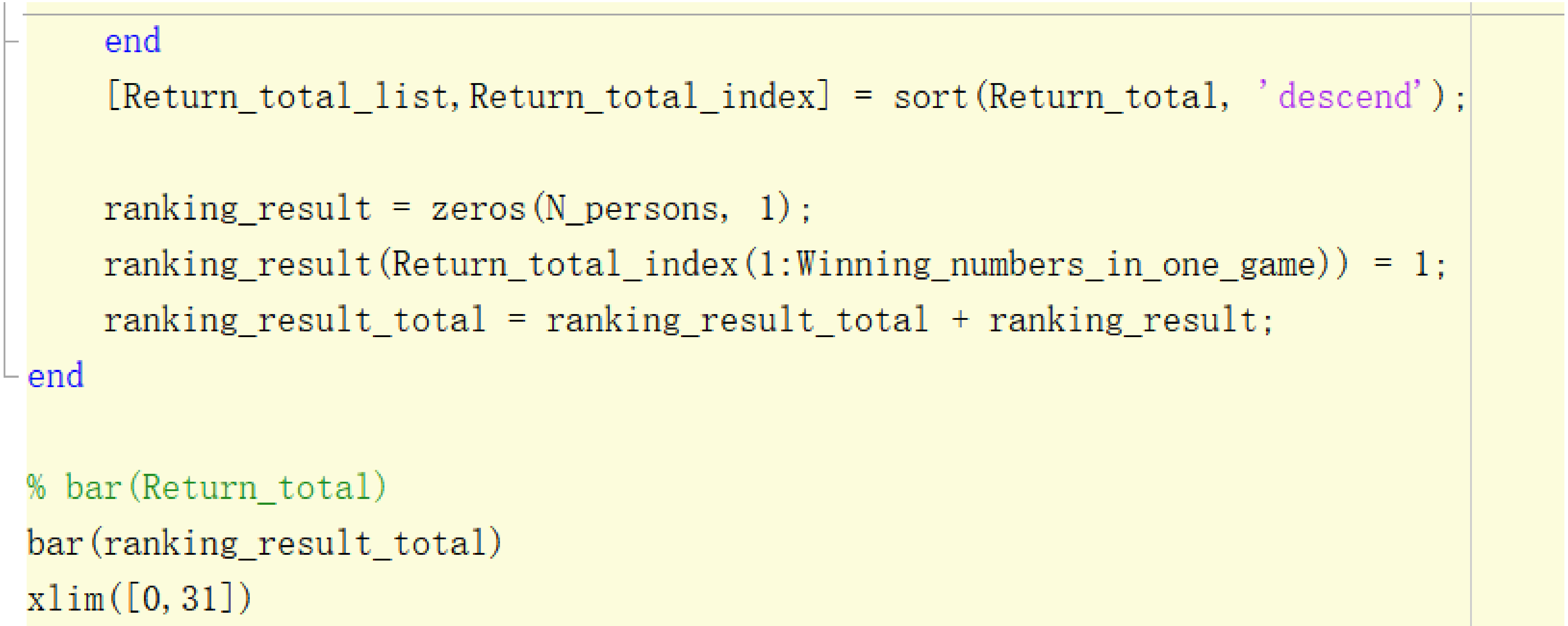
- In each game, there are totally 20 trades. That is more realistic as one person cannot perform 5000 ‘important’ trades in his whole life
- In each game, take the top 5 person with highest value as winners, and record them with ‘+1’
- Repeat the game 1000 times.
    - This is the basic idea of ‘Statistics’, and also exactly the idea of ‘Monte Carlo’ Simulation: repeat an experiment a large number of times, and output the average result
    - It means: if you can repeat your life 1000 times, what is the total numbers you can win, or
    - The average performance of your strategy in 1000 parallel worlds

Results from Last Game



```
clear;
clc;
% the above two commands clear all the previous record in the Memory

Repeated_trails = 1000;                 % repeat the game Repeated_trails times
N_trades = 20;                     % trade N_trades times
N_persons = 30;                     % totally N_persons persons
N_persons_pairs = round(N_persons/2);      % N_persons_pairs groups

X_betray_trust_point = 4;
Y_betray_betray_point = 5;

Winning_numbers_in_one_game = 5;          % assume that the first 5 persons win in this gam
ranking_result_total = zeros(N_persons, 1); % store the result of all the games
```

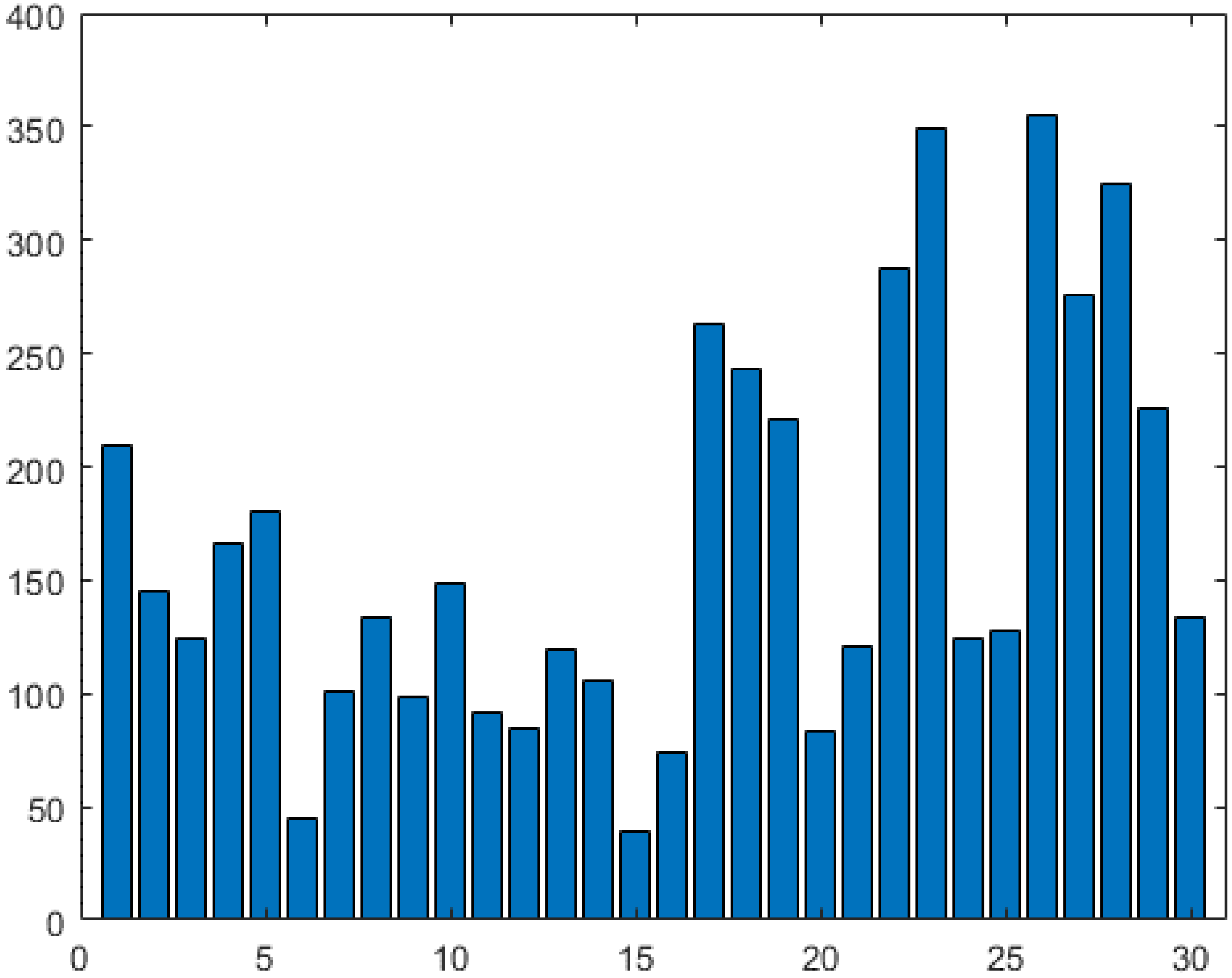
Results from Last Game



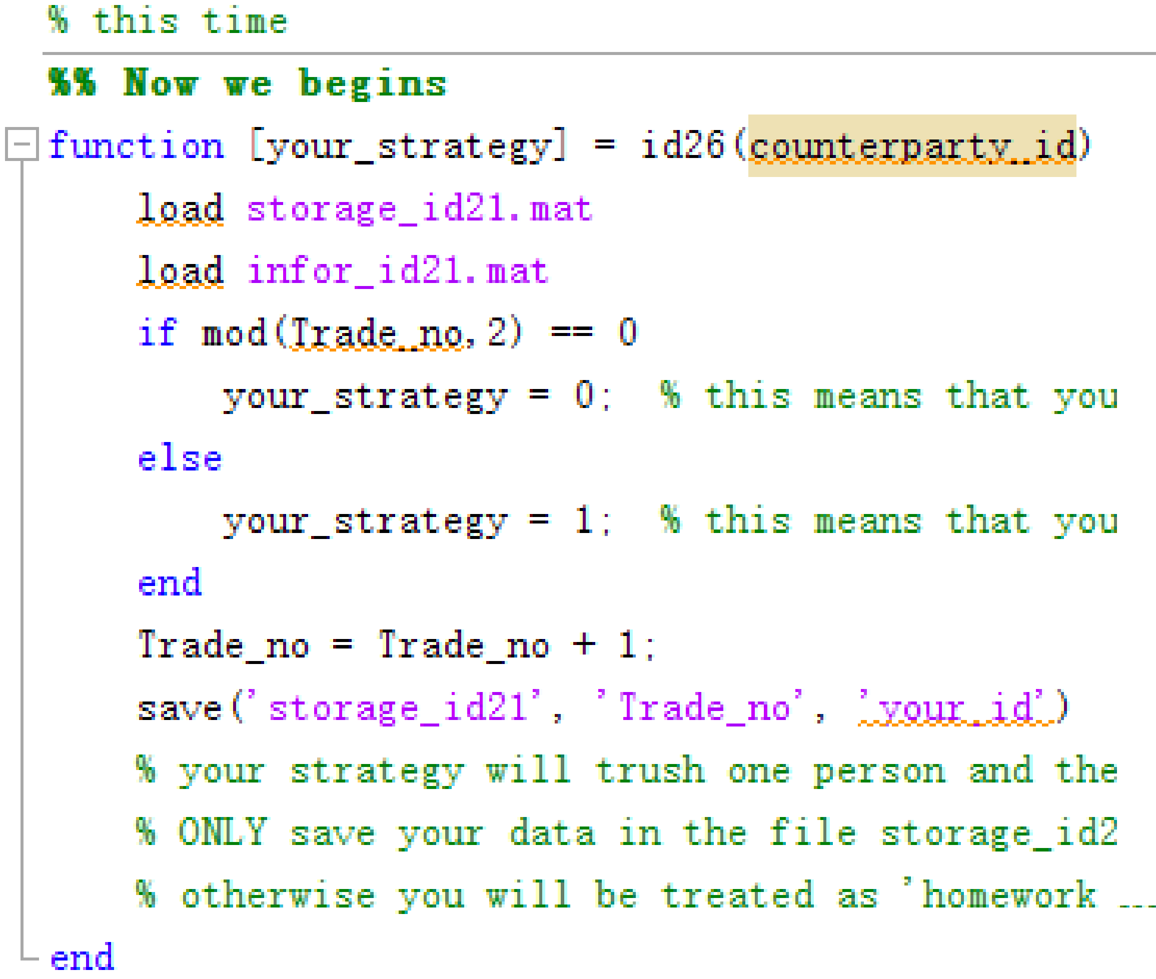
```
    end
    [Return_total_list,Return_total_index] = sort(Return_total, 'descend');

    ranking_result = zeros(N_persons, 1);
    ranking_result(Return_total_index(1:Winning_numbers_in_one_game)) = 1;
    ranking_result_total = ranking_result_total + ranking_result;
end

% bar(Return_total)
bar(ranking_result_total)
xlim([0,31])
```

Results from Last Game

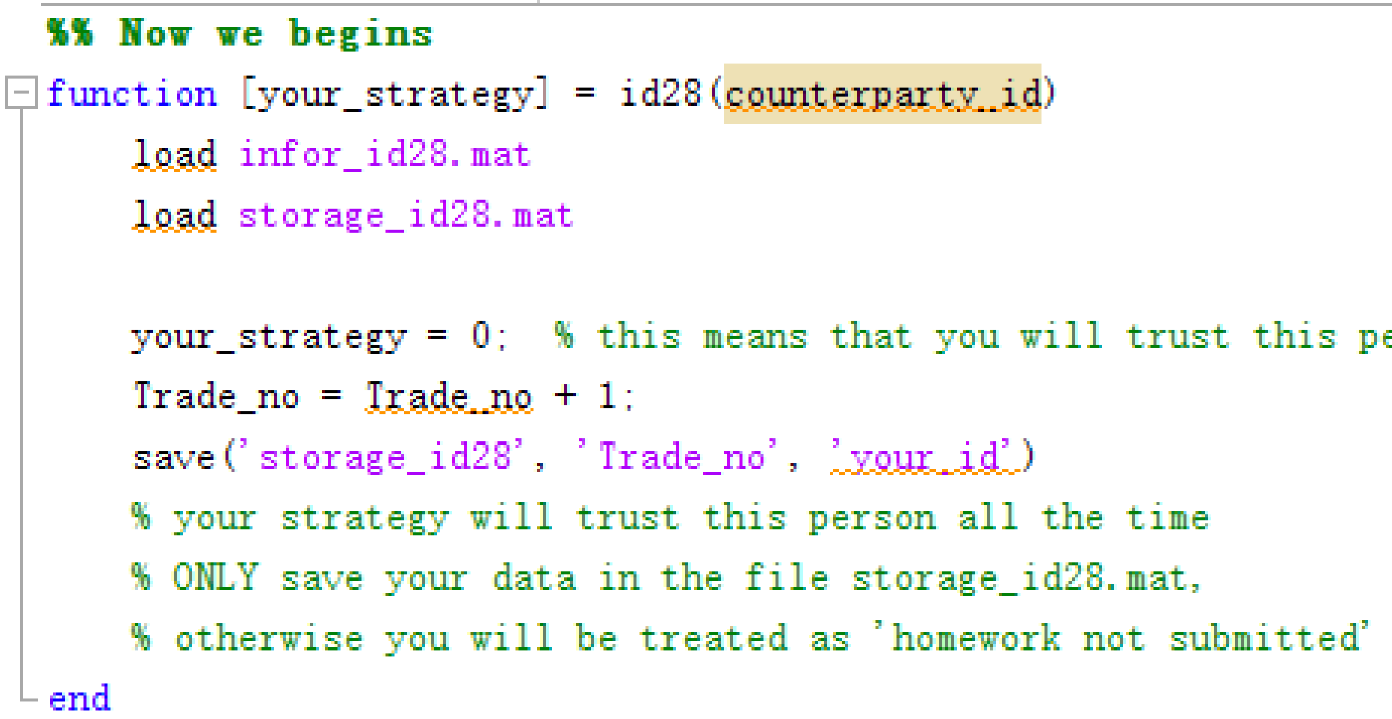
- Highest 5:
  - ID22
  - ID23
  - ID26
  - ID27
  - ID28



Results from Last Game



```
% this time
%% Now we begins
function [your_strategy] = id26(counterparty_id)
    load storage_id21.mat
    load infor_id21.mat
    if mod(Trade_no,2) == 0
        your_strategy = 0;  % this means that you
    else
        your_strategy = 1;  % this means that you
    end
    Trade_no = Trade_no + 1;
    save('storage_id21', 'Trade_no', 'your_id')
    % your strategy will trush one person and the
    % ONLY save your data in the file storage_id2
    % otherwise you will be treated as 'homework
end
```



```
%% Now we begins
function [your_strategy] = id28(counterparty_id)
    load infor_id28.mat
    load storage_id28.mat

    your_strategy = 0;  % this means that you will trust this pe
    Trade_no = Trade_no + 1;
    save('storage_id28', 'Trade_no', 'your_id')
    % your strategy will trust this person all the time
    % ONLY save your data in the file storage_id28.mat,
    % otherwise you will be treated as 'homework not submitted'
end
```

Results from Last Game

- Interesting conclusion:
  - In fact, winners will trust a lot, you can test the program yourself
  - Luck is very important, although some other IDs are exactly the same, the results are slightly different
  - But it might be not real for real world, therefore, we need to modify the game!

Results from Last Game

- The deficiencies of the ‘ranking’ index:
  - In 1000 games, if one strategy win 800 times with final value 10, but loss 200 times with final value -50, it will be given a high score in this ‘ranking’ index
  - But this strategy is very stupid: the expectation is 800*10-50*200=-2000!
  - We can use the ‘average return’ and the ‘ranking’ to evaluate the strategy comprehensively!

  - You can re-write this program by yourself.

Experiment: changing a simulation system

- Now consider this updated game:
  1. There are N=2K persons in the game
  2. In each time, two person (K pairs) will trade with each other
  3. You and your counterparties both have three options:
     - Trade, or says, trust
     - Cheat, or says, betray
     - Reject, or says, refuse to trade

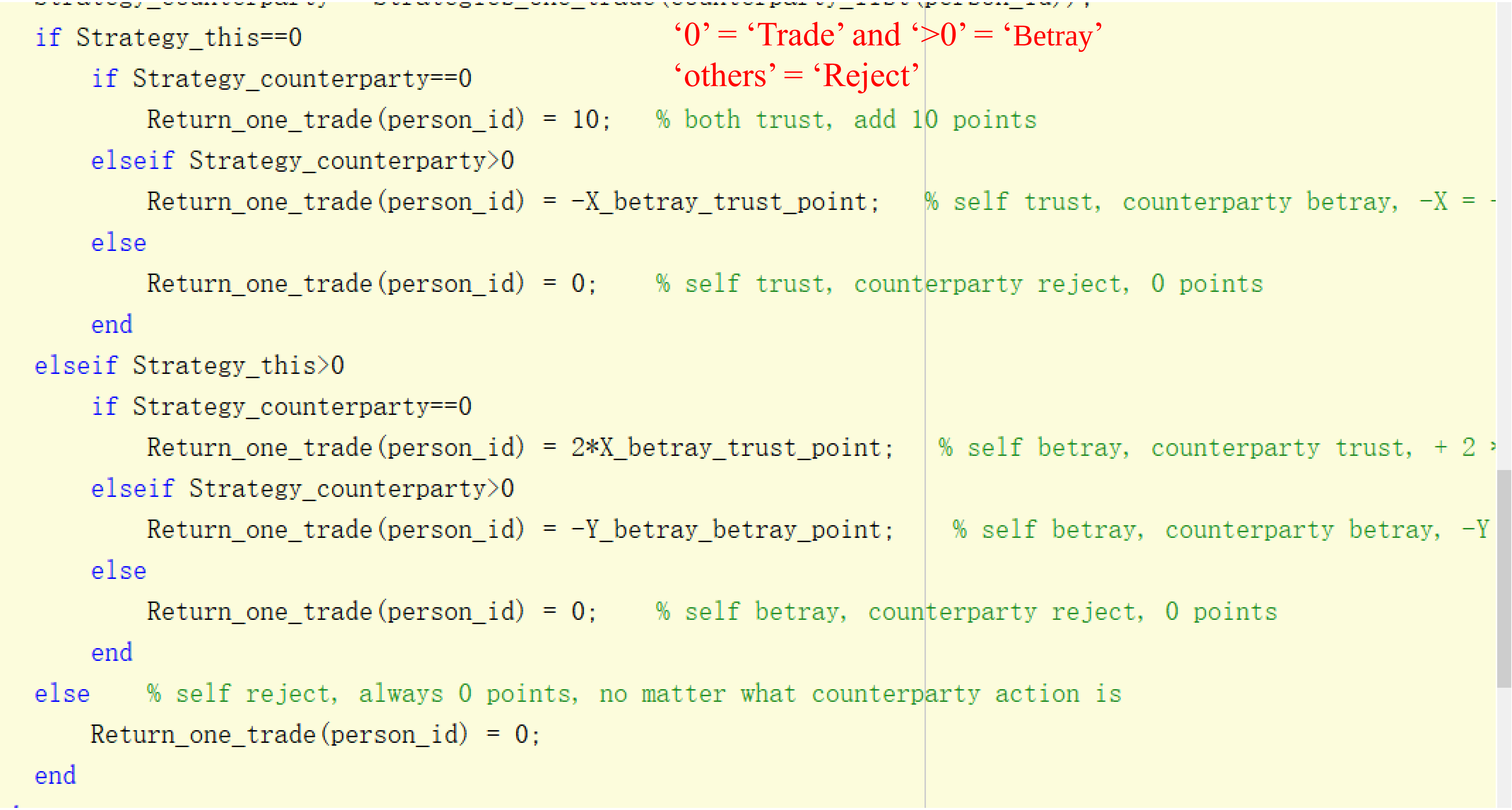
     Once both persons choose his/her options, calculate the points he/she get as the table

		A		
		Trust	Betray	Reject
B	Trust	A: +10; B: +10	A: +2X; B: -X	A&B: +0
	Betray	A: -X; B: +2X	A: -Y; B: -Y	A&B: +0
	Reject	A&B: +0	A&B: +0	A&B: +0

Experiment: changing a simulation system

- Now, you are not the participant only, you are the game designer!
- You need to consider two parts:
  - Technique part: how to change the system so that the ‘Reject’ can be added?
  - strategy part: how to design X and Y so that the ‘Reject’ is useful?
- The Technique part is easy:
  - In previous system, we design ‘0’ = ‘Trade’ and ‘not 0’ = ‘Betray’
  - Quick question: why not assign ‘0’ = ‘Trade’ and ‘1’ = ‘Betray’?
    - Because, in that case, one might give a strategy ‘2’, or ‘A’, and the system will fails.
  - Therefore, we must define at least one option of the system as ‘others’
  - For example:
    - ‘0’ = ‘Trade’ and ‘>0’ = ‘Betray’
    - ‘others’ = ‘Reject’

Experiment: changing a simulation system



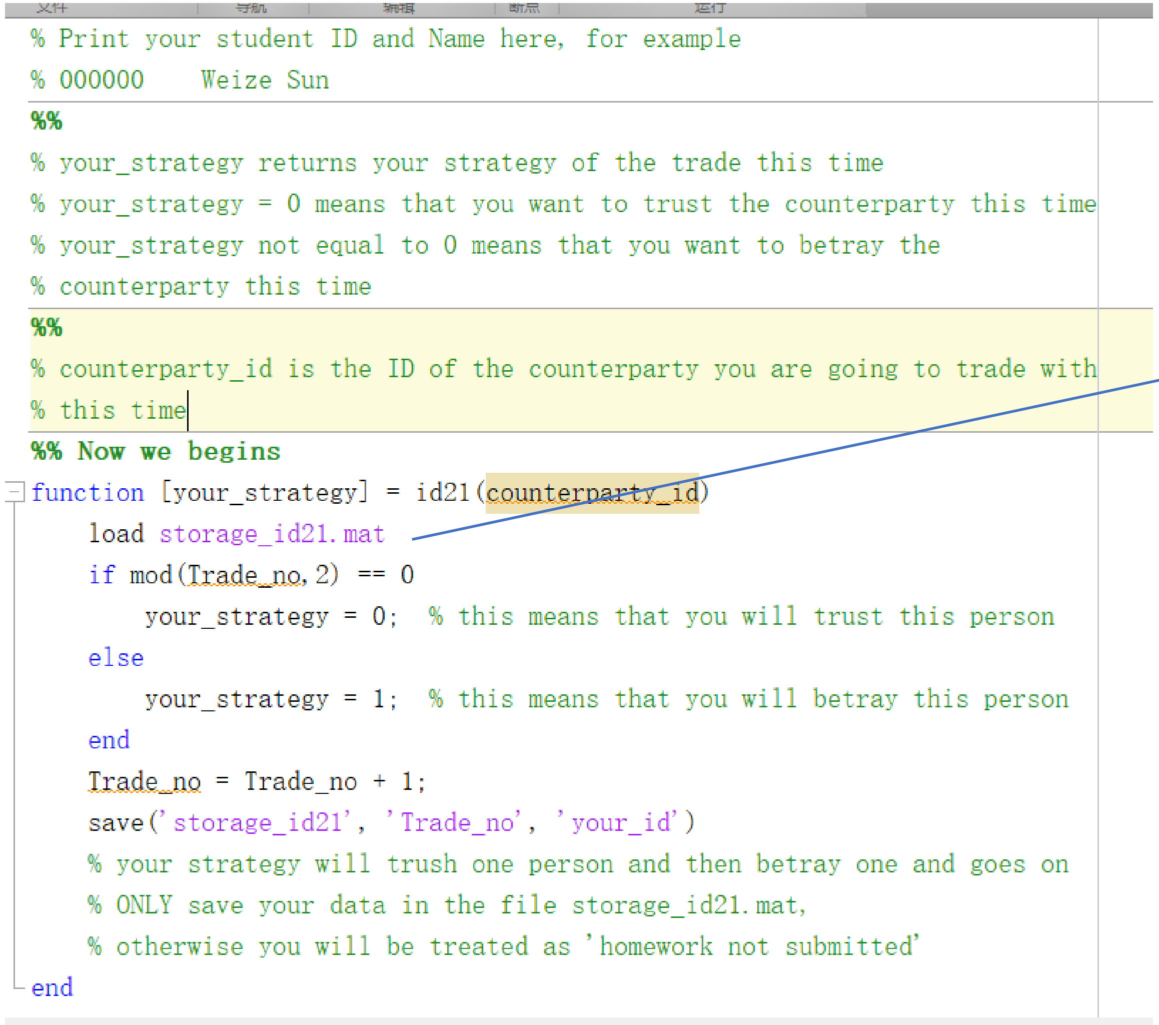
‘0’ = ‘Trade’ and ‘>0’ = ‘Betray’

‘others’ = ‘Reject’

```
Strategy_counterparty = Strategies_one_trade(counterparty_list(person_id));
if Strategy_this==0
    if Strategy_counterparty==0
        Return_one_trade(person_id) = 10;   % both trust， add 10 points
    elseif Strategy_counterparty>0
        Return_one_trade(person_id) = -X_betray_trust_point;   % self trust， counterparty betray， -X = -
    else
        Return_one_trade(person_id) = 0;    % self trust， counterparty reject， 0 points
    end
elseif Strategy_this>0
    if Strategy_counterparty==0
        Return_one_trade(person_id) = 2*X_betray_trust_point;   % self betray， counterparty trust， + 2 *
    elseif Strategy_counterparty>0
        Return_one_trade(person_id) = -Y_betray_betray_point;    % self betray， counterparty betray， -Y
    else
        Return_one_trade(person_id) = 0;    % self betray， counterparty reject， 0 points
    end
else    % self reject， always 0 points， no matter what counterparty action is
    Return_one_trade(person_id) = 0;
end
```

Experiment: changing a simulation system

- Before we goes on, let’s see some bugs, or the unsatisfying parts of the previous system, and change them first.
- Let’s take the previous default id21.m as an example



```
% Print your student ID and Name here, for example
% 000000    Weize Sun
%%
% your_strategy returns your strategy of the trade this time
% your_strategy = 0 means that you want to trust the counterparty this time
% your_strategy not equal to 0 means that you want to betray the
% counterparty this time
%%
% counterparty_id is the ID of the counterparty you are going to trade with
% this time
%% Now we begins
function [your_strategy] = id21(counterparty_id)
    load storage_id21.mat
    if mod(Trade_no, 2) == 0
        your_strategy = 0;  % this means that you will trust this person
    else
        your_strategy = 1;  % this means that you will betray this person
    end
    Trade_no = Trade_no + 1;
    save('storage_id21', 'Trade_no', 'your_id')
    % your strategy will trush one person and then betray one and goes on
    % ONLY save your data in the file storage_id21.mat,
    % otherwise you will be treated as 'homework not submitted'
end
```

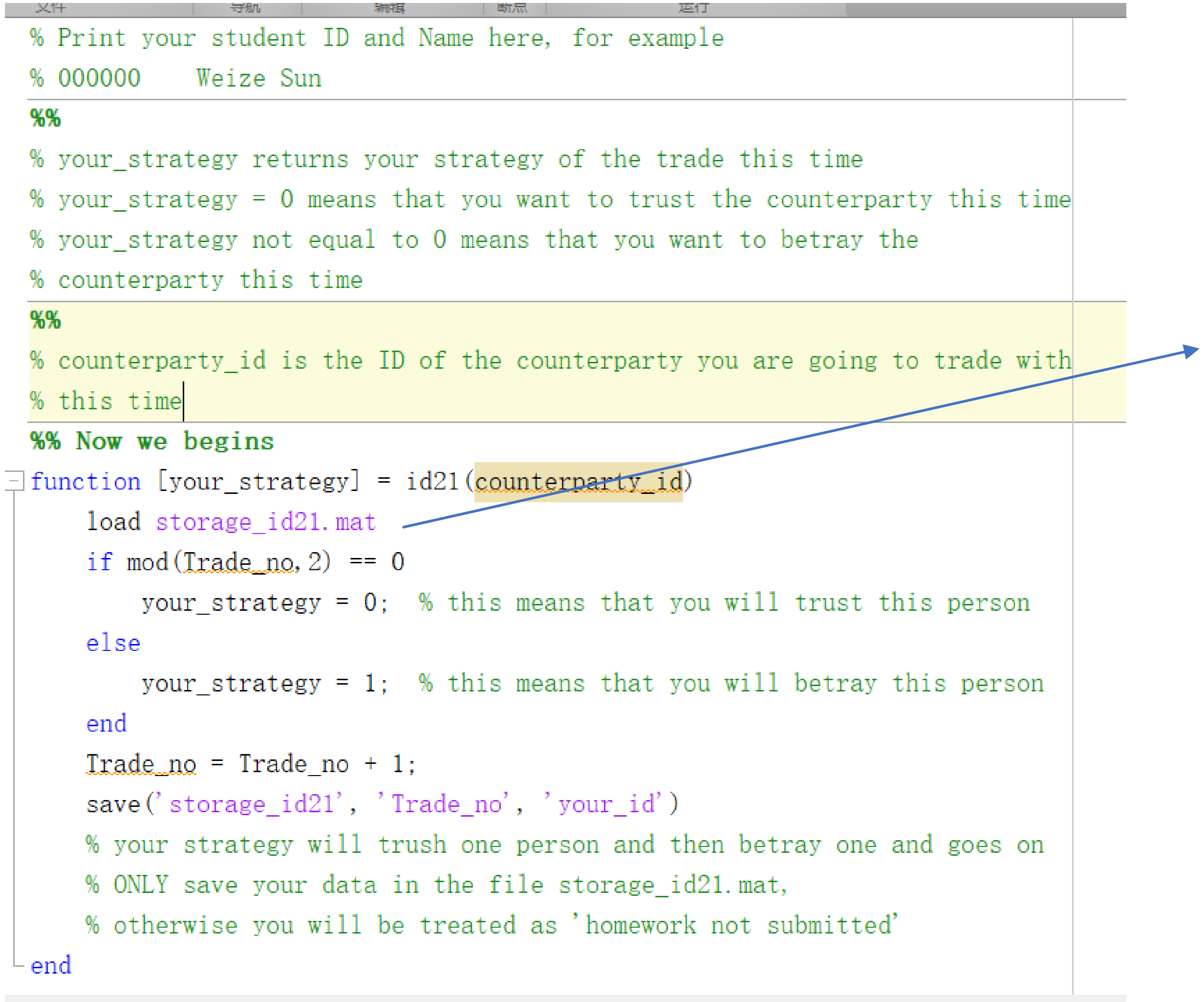
Here, it load the ‘storage_id21.mat’ at default, and use the ‘Trade_no’ to decide his strategy. It will leads to the following problems:

1. If there is no ‘storage_id21.mat’, the system will fails.
2. The ‘storage_id21.mat’ might give a default value of ‘Trade_no’ randomly, making the strategy very unstable

- For a game participant, he should care about the 2nd problem majorly (also, he should care about the 1st problem in order to avoid failure of his program);
- But, for the game designer, he should care about the 1st problem seriously!

Experiment: changing a simulation system

- Before we goes on, let’s see some bugs, or the unsatisfying parts of the previous system, and change them first.
- Let’s take the previous default id21.m as an example

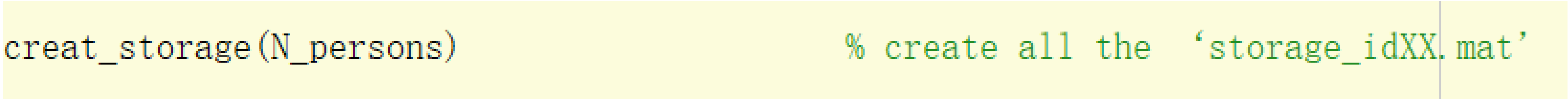


```
% Print your student ID and Name here, for example
% 000000    Weize Sun
%%
% your_strategy returns your strategy of the trade this time
% your_strategy = 0 means that you want to trust the counterparty this time
% your_strategy not equal to 0 means that you want to betray the
% counterparty this time
%%
% counterparty_id is the ID of the counterparty you are going to trade with
% this time
%% Now we begins
function [your_strategy] = id21(counterparty_id)
    load storage_id21.mat
    if mod(Trade_no,2) == 0
        your_strategy = 0;  % this means that you will trust this person
    else
        your_strategy = 1;  % this means that you will betray this person
    end
    Trade_no = Trade_no + 1;
    save('storage_id21', 'Trade_no', 'your_id')
    % your strategy will trush one person and then betray one and goes on
    % ONLY save your data in the file storage_id21.mat,
    % otherwise you will be treated as 'homework not submitted'
end
```

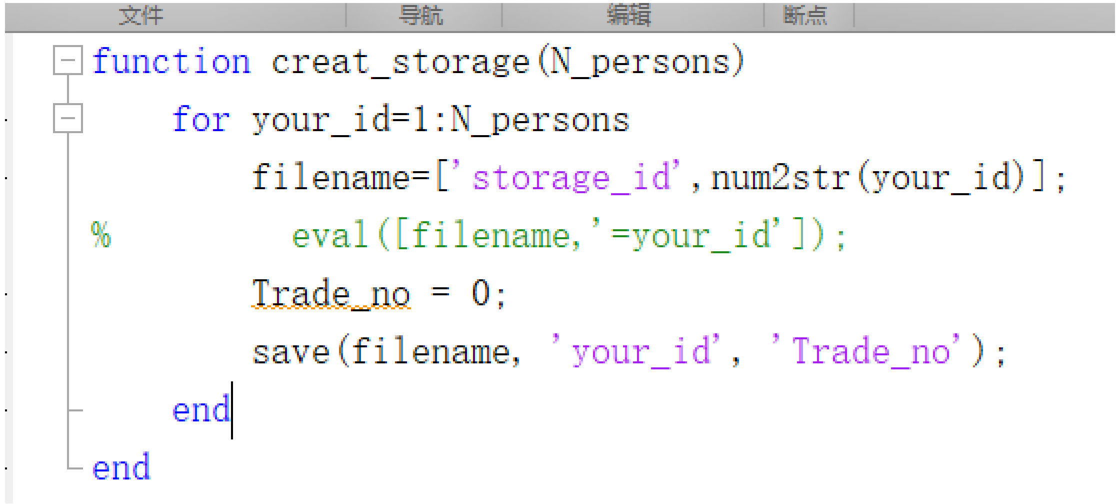
Therefore, in the main program, we should add some codes to generate a default ‘storage_idXX.mat’ with a parameter ‘Trade_no’, then the system will be of less possibility to fail.

Experiment: changing a simulation system

- That is:



```
creat_storage(N_persons)                    % create all the  ‘storage_idXX.mat’
```



```
function creat_storage(N_persons)
    for your_id=1:N_persons
        filename=['storage_id',num2str(your_id)];
%         eval([filename,'=your_id']);
        Trade_no = 0;
        save(filename, 'your_id', 'Trade_no');
    end
end
```

- 这段程序的作用为：在主程序中，在游戏开始时，给所有的storage_idXX.mat文件里面创造了一个Trade_no=0，这等于给了一个约束和指示：
  - 约束：使每次游戏开始时，有一个统一的标准数字
  - 指示：作为游戏的参与者（IDXX），参与者不能调用主程序的数据，但是可以看到自己的storage_idXX.mat文件中的Trade_no；当它为0的时候，说明游戏刚开始。而这个Trade_no可以在每次游戏后+1，指示游戏的进度。

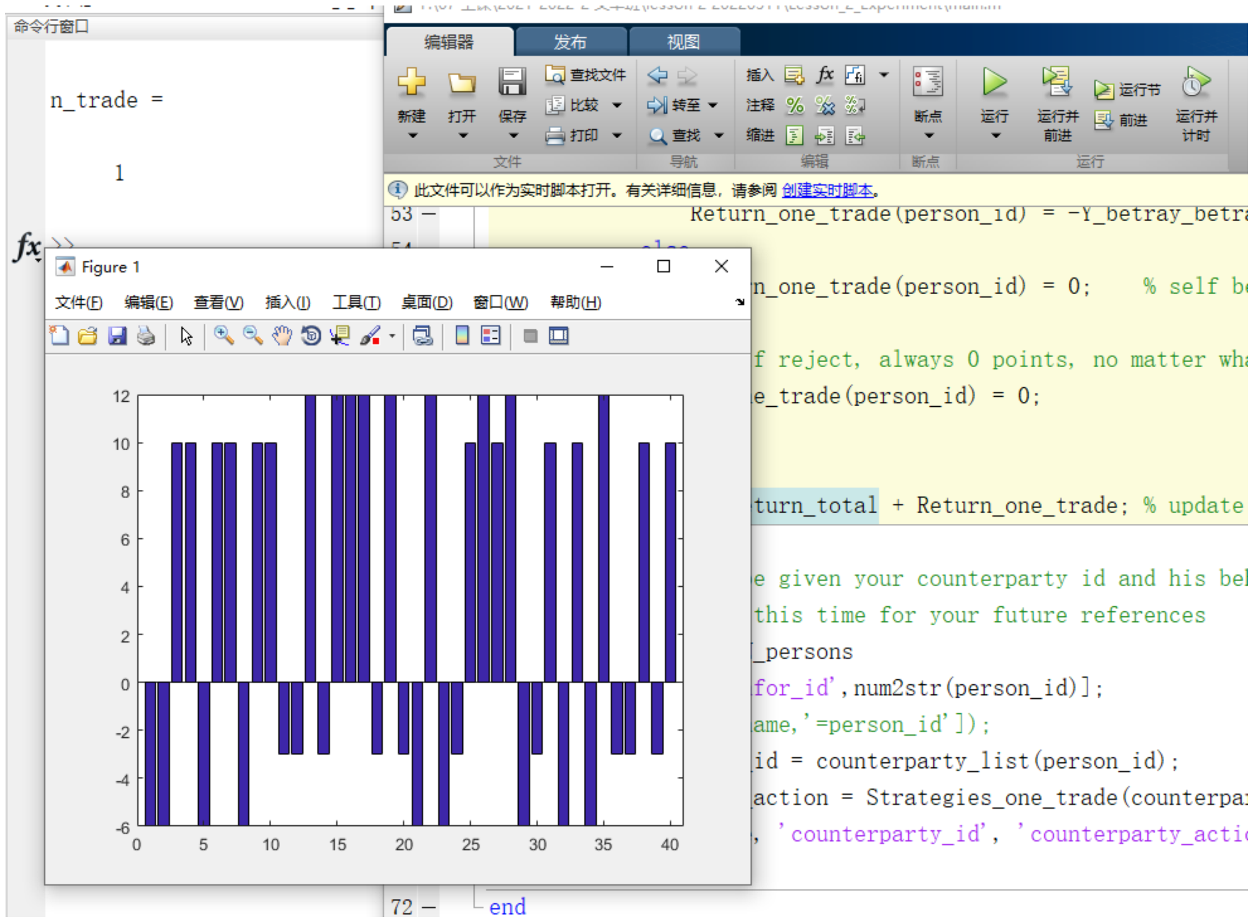
Experiment: changing a simulation system

- Similarly, you are also required to take care of the ‘infor_idXX.mat’, as its generation code is placed in the end of the main loop.
- If there are no ‘infor_idXX.mat’, for example, ‘infor_id31.mat’, in the very beginning, the system will fail!

- Try to modify this system by yourself

Experiment: changing a simulation system

- For example: there is ‘infor_idXX.mat’



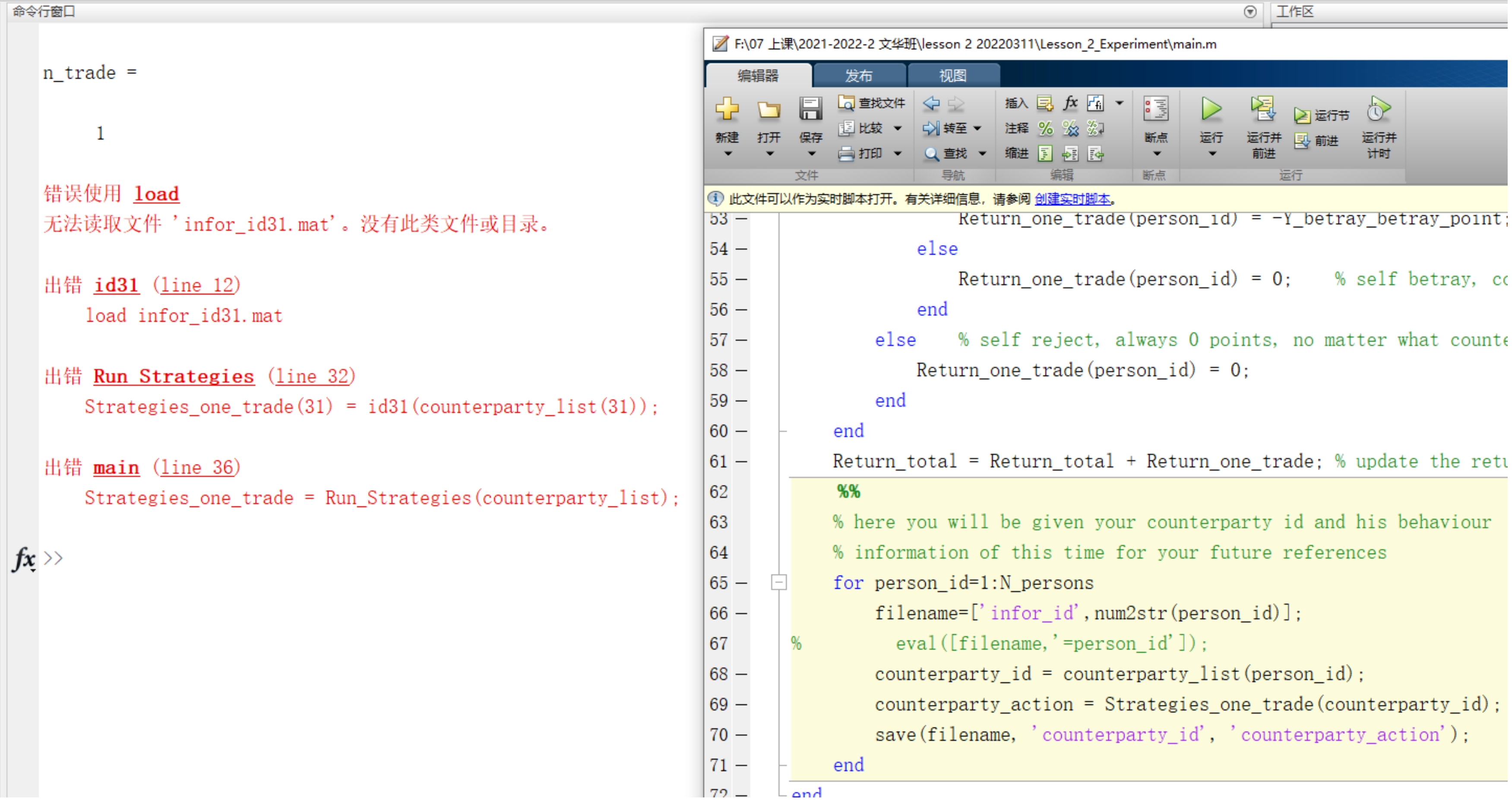
```
n_trade =

     1

>>
```

ⓘ 此文件可以作为实时脚本打开。有关详细信息，请参阅 创建实时脚本。

```
            Return_one_trade(person_id) = -Y_betray_betra
n_one_trade(person_id) = 0;     % self be
f reject, always 0 points, no matter wha
e_trade(person_id) = 0;
turn_total + Return_one_trade; % update
e given your counterparty id and his bel
this time for your future references
_persons
for_id',num2str(person_id)];
ame,'=person_id']);
id = counterparty_list(person_id);
action = Strategies_one_trade(counterpa
, 'counterparty_id', 'counterparty_acti
end
```

Experiment: changing a simulation system

- For example: no ‘infor_idXX.mat’



```
n_trade =

     1

错误使用 load
无法读取文件 'infor_id31.mat'。没有此类文件或目录。

出错 id31 (line 12)
    load infor_id31.mat

出错 Run_Strategies (line 32)
    Strategies_one_trade(31) = id31(counterparty_list(31));

出错 main (line 36)
    Strategies_one_trade = Run_Strategies(counterparty_list);

fx >>
```

F:\07 上课\2021-2022-2 文华班\lesson 2 20220311\Lesson_2_Experiment\main.m

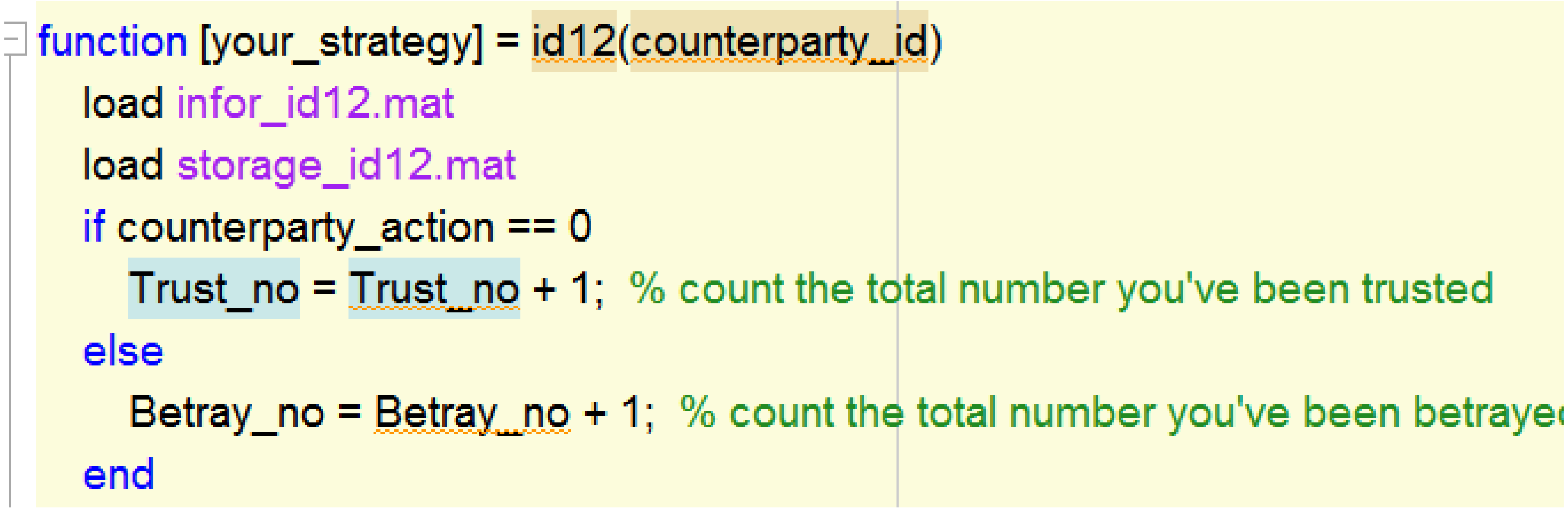
此文件可以作为实时脚本打开。有关详细信息，请参阅 创建实时脚本。

```
                Return_one_trade(person_id) = -Y_betray_betray_point;
            else
                Return_one_trade(person_id) = 0;     % self betray, cc
            end
        else     % self reject, always 0 points, no matter what counte
            Return_one_trade(person_id) = 0;
        end
    end
    Return_total = Return_total + Return_one_trade; % update the retu
    %%
    % here you will be given your counterparty id and his behaviour
    % information of this time for your future references
    for person_id=1:N_persons
        filename=['infor_id',num2str(person_id)];
%         eval([filename,'=person_id']);
        counterparty_id = counterparty_list(person_id);
        counterparty_action = Strategies_one_trade(counterparty_id);
        save(filename, 'counterparty_id', 'counterparty_action');
    end
end
```

- Try to modify this system by yourself

Experiment: changing a simulation system

- Now there comes another problem. Using the ID12 from last year as example:



```
function [your_strategy] = id12(counterparty_id)
  load infor_id12.mat
  load storage_id12.mat
  if counterparty_action == 0
     Trust_no = Trust_no + 1;  % count the total number you've been trusted
  else
     Betray_no = Betray_no + 1;  % count the total number you've been betraye
  end
```

if ‘Trust_no’ / ‘Betray_no’ does not exists, it will fails

However, as the ‘infor_idXX.mat’ are new generated, ‘Trust_no’ / ‘Betray_no’ does not exists

- How to modify?

Experiment: changing a simulation system

- Now there comes another problem. Using the ID12 from last year as example:

if ‘Trust_no’ / ‘Betray_no’ does not exists, it will fails

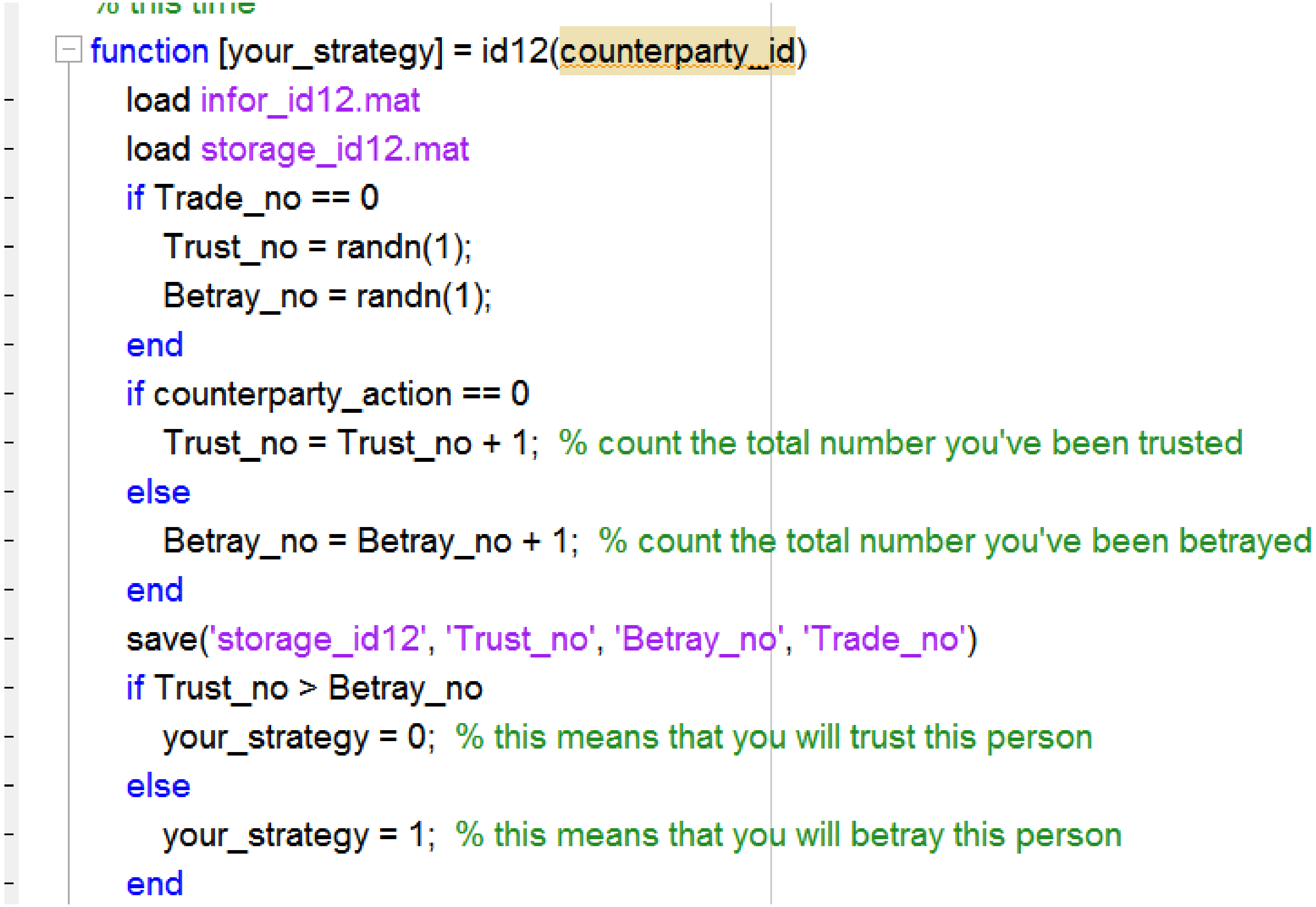
However, as the ‘infor_idXX.mat’ are new generated, ‘Trust_no’ / ‘Betray_no’ does not exists

- How to modify?

- 这段程序的作用为：在主程序中，在游戏开始时，给所有的storage_idXX.mat文件里面创造了一个Trade_no=0，这等于给了一个约束和指示：
  - 约束：使每次游戏开始时，有一个统一的标准数字
  - 指示：作为游戏的参与者（IDXX），参与者不能调用主程序的数据，但是可以看到自己的storage_idXX.mat文件中的Trade_no；当它为0的时候，说明游戏刚开始。而这个Trade_no可以在每次游戏后+1，指示游戏的进度。

Experiment: changing a simulation system

- Now there comes another problem. Using the ID12 from last year as example:



```
% this time
function [your_strategy] = id12(counterparty_id)
    load infor_id12.mat
    load storage_id12.mat
    if Trade_no == 0
        Trust_no = randn(1);
        Betray_no = randn(1);
    end
    if counterparty_action == 0
        Trust_no = Trust_no + 1;  % count the total number you've been trusted
    else
        Betray_no = Betray_no + 1;  % count the total number you've been betrayed
    end
    save('storage_id12', 'Trust_no', 'Betray_no', 'Trade_no')
    if Trust_no > Betray_no
        your_strategy = 0;  % this means that you will trust this person
    else
        your_strategy = 1;  % this means that you will betray this person
    end
```

- Modify your program accordingly

Experiment: changing a simulation system

- Now we go back to the strategy problem: how to design X and Y so that the ‘Reject’ is useful?
- Generally speaking, it is a ‘strategy’ or ‘Sociology(社会学)’ problem
  - Different person can give different ideology, thus leading to different X and Y
- But, this is a ~~math~~ programming course!
- Therefore, I will simply introduce a ‘programming’ method to decide the X and Y.

Experiment: changing a simulation system – the programming method

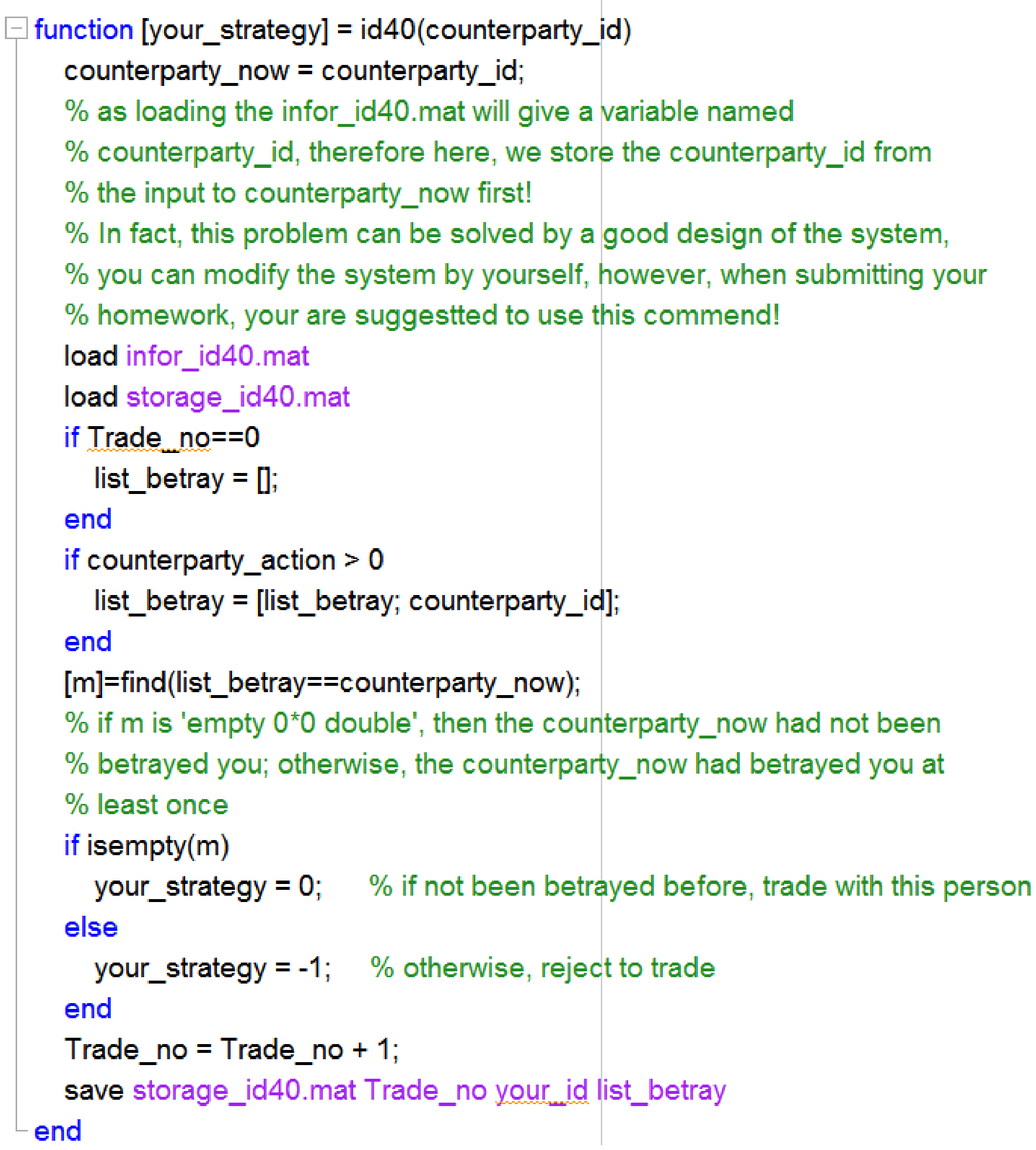
- We begin with the following assumptions:
  - The previous default 30 strategies, i.e., id1 to id30, are appropriate
    - 10 always trade; 10 always betray; 10 trade once and then betray once and then go on
  - There are 10 more strategies for us to test the ‘reject’ option
    - Of course, you can use more, for example, you can set ‘N_persons = 300’ and test 270 strategies to see how ‘reject’ works, but that is another story.
  - For these 10 strategies, if the ‘reject’ option is not chosen, randomly trade or betray
    - Other strategies can be set, but this ‘randomly trade or betray’ can make the system more robust to the general case: there are always trade, and betray

Experiment: changing a simulation system – the programming method

- How to design?
  - 10 strategies, with ‘reject’ probability 10% to 100%
  - trade 1000 times
  - Test several pairs of X and Y values, and see which pair make the strategy that ‘reject’ with probability 30% wins
  - Choose this X and Y

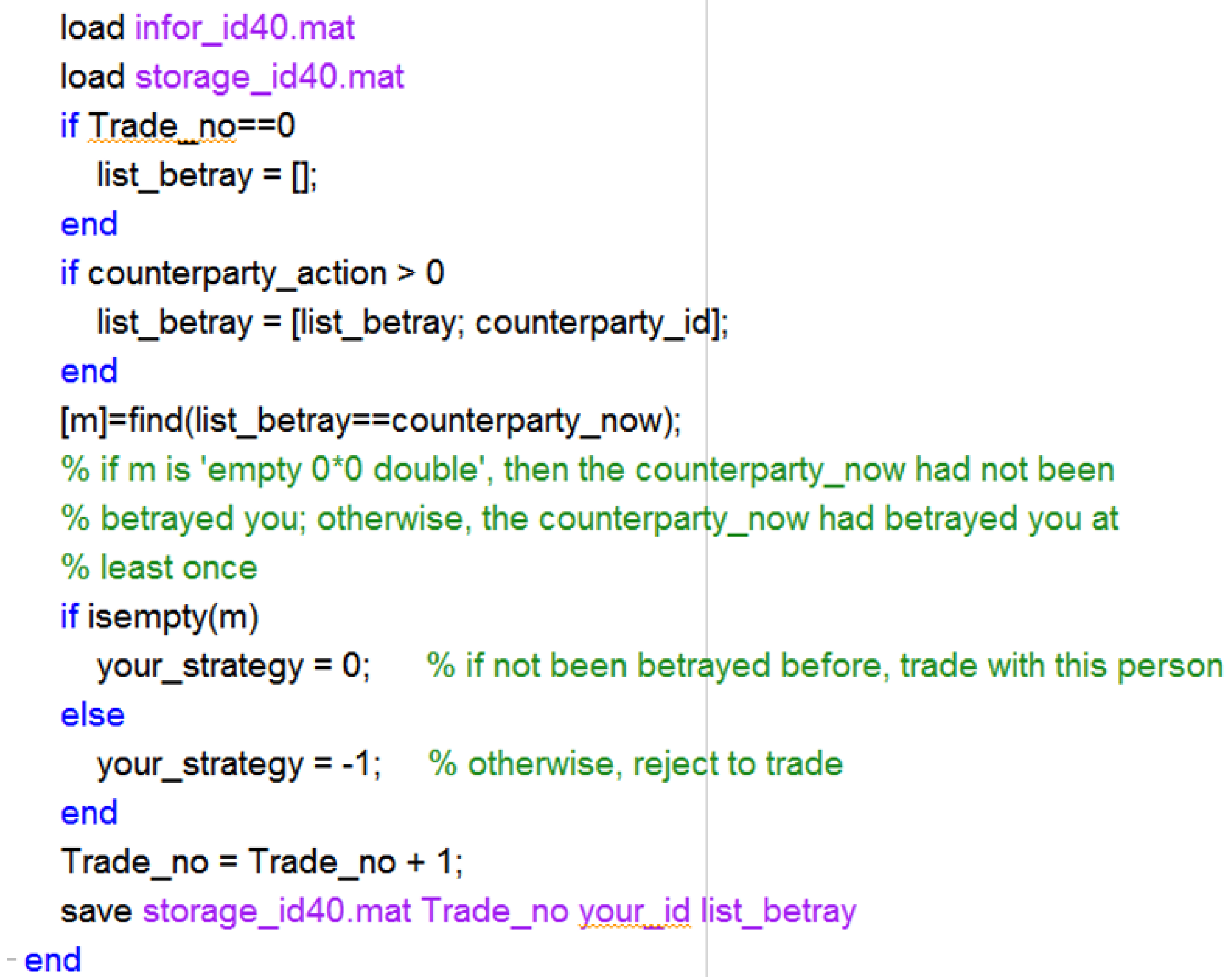
- Note: this is just one idea, you can use your idea to do the design
- Your are encourage to try this test, but it is not a homework
- Here I will show one ‘reject after being betrayed’ strategy

Experiment: the reject after being betrayed strategy



```
function [your_strategy] = id40(counterparty_id)
    counterparty_now = counterparty_id;
    % as loading the infor_id40.mat will give a variable named
    % counterparty_id, therefore here, we store the counterparty_id from
    % the input to counterparty_now first!
    % In fact, this problem can be solved by a good design of the system,
    % you can modify the system by yourself, however, when submitting your
    % homework, your are suggestted to use this commend!
    load infor_id40.mat
    load storage_id40.mat
    if Trade_no==0
        list_betray = [];
    end
    if counterparty_action > 0
        list_betray = [list_betray; counterparty_id];
    end
    [m]=find(list_betray==counterparty_now);
    % if m is 'empty 0*0 double', then the counterparty_now had not been
    % betrayed you; otherwise, the counterparty_now had betrayed you at
    % least once
    if isempty(m)
        your_strategy = 0;     % if not been betrayed before, trade with this person
    else
        your_strategy = -1;    % otherwise, reject to trade
    end
    Trade_no = Trade_no + 1;
    save storage_id40.mat Trade_no your_id list_betray
end
```

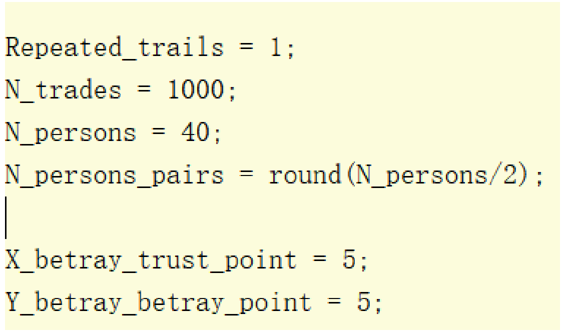
Experiment: the reject after being betrayed strategy



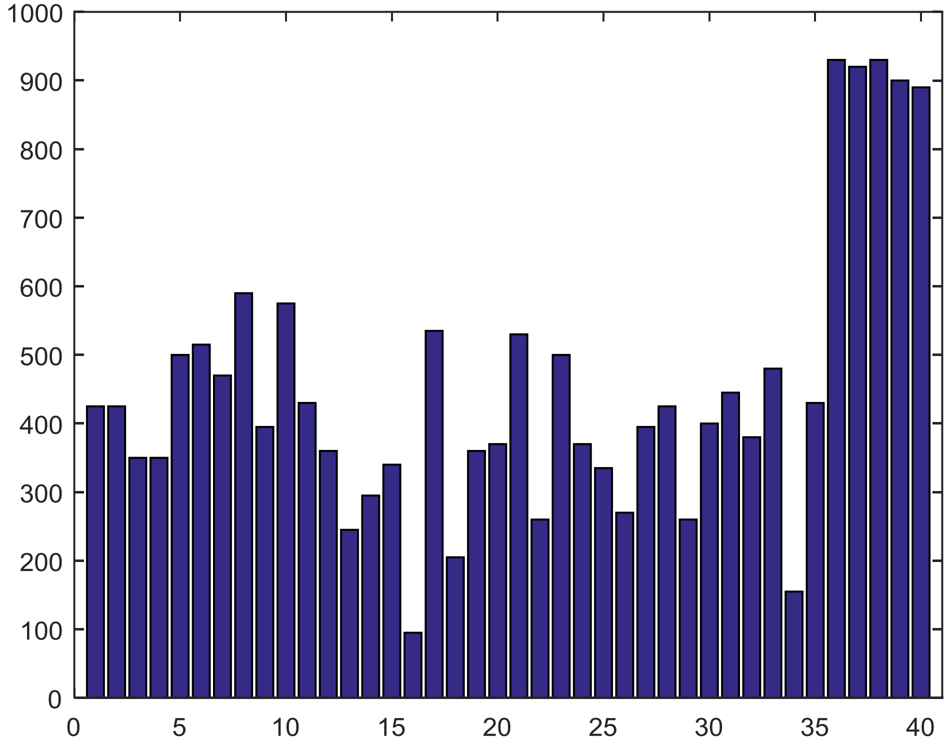
```
    load infor_id40.mat
    load storage_id40.mat
    if Trade_no==0
        list_betray = [];
    end
    if counterparty_action > 0
        list_betray = [list_betray; counterparty_id];
    end
    [m]=find(list_betray==counterparty_now);
    % if m is 'empty 0*0 double', then the counterparty_now had not been
    % betrayed you; otherwise, the counterparty_now had betrayed you at
    % least once
    if isempty(m)
        your_strategy = 0;     % if not been betrayed before, trade with this person
    else
        your_strategy = -1;    % otherwise, reject to trade
    end
    Trade_no = Trade_no + 1;
    save storage_id40.mat Trade_no your_id list_betray
end
```

Experiment: the result

- The settings
  - The setting now: {X=5 and Y=5}, and {Repeated_trails=1, N_trades=200}
  - 10 more default strategies given: 5 ‘betray always’ and 5 ‘reject after being betrayed’, named as id31-35 and 36-40
  - for id1-30, we use default in as previous lesson
  - Therefore, there are now 40 persons



```
Repeated_trails = 1;
N_trades = 1000;
N_persons = 40;
N_persons_pairs = round(N_persons/2);

X_betray_trust_point = 5;
Y_betray_betray_point = 5;
```



- It is not strange that id36-40 wins, as I design the ‘reject after being betrayed’ strategy based on the fact that many players betray a lot
- I hope that you can beat my ‘reject after being betrayed’ strategy under {X=5 and Y=5} and {X=8 and Y=5}, in this homework
- Note that, a strategy always trade will not as good as enough (but usable) this time

Experiment: homework

- Programming homework today
  - The setting now: {X=5 and Y=5}/{X=8 and Y=5}, and {Repeated_trails=1, N_trades=1000}
  - Default strategies: 5 ‘betray always’ and 5 ‘reject after being betrayed’
  - Add the option ‘reject’
  - You can redesign your strategy, and submit it before 03.15 (Fir.) 23:59:59, to 译哲’s email box
  - The problems listed today must not happen again!

  - The first two homework (last week and this week) will be attached as attachments of mini project / experimental report 1, with approx. 20-30 points
  - Homework submitted in time, and **runnable**, will get all these points

ID列表

- 请严格按照本列表，交自己ID号的程序
- 23:59:59, 15/03前直接email交给刘译哲

交程序的ID号	学号	姓名
1	2022040399	张桂嘉
2	2022090123	徐雷
3	2022110131	廖祖颐
4	2022270054	詹兴足
5	2022280039	郭瑞煜
6	2022280069	曾颖岚
7	2022280105	郑志锰
8	2022280142	崔殷霖
9	2022280160	姚宇铭
10	2022280162	曾源原
11	2022280179	李梓琦
12	2022280247	林茵茵
13	2022280297	陈应权
14	2022280307	叶朗钊
15	2022280327	古炜
16	2022280365	郭展鹏
17	2022280380	杨烨
18	2022280419	薛玉龙
19	2022280432	彭佳
20	2022280445	何雨璇
21	2022280450	孙浩然
22	2022280453	陈奇峰
23	2022280485	贾苏健
24	2022280546	张梓荣
25	2022280553	林凡超
26	2022280562	陈柯瑜
27	2022280573	王梓为
28	2022280574	马海洲
29	2022300013	卫宏林