Table of Contents

JIA Jiyuan 20210122 ICE#2 Class 01 1
Problem 1: 1
Problem 2: 1
Problem 3: 2
Problem 4: 3
Problem 5: 3
Problem 6: 4
Problem 7: 4
Problem 8: 5
Attachment of function 6

JIA Jiyuan 20210122 ICE#2 Class 01

```
clear;clc;
```

Problem 1:

returns a matrix that contains only those elements of M that are in even rows and columns

```
clear;clc;
% random matrix
M = 100*rand(7,7)
% give elements in even rows and columns
even_index(M)
```

```
M =

   40.2970   57.6776   32.2482   32.7024   70.8671   70.3953   40.7619
   10.7040   94.4029   97.6147   80.4069   23.4926   93.2303   36.8700
   72.4166   87.1452   27.8211   53.8250   39.8896   68.7653   46.8399
   61.3682   50.7602    7.2831   46.3295   26.8124   56.8354   50.3414
   78.2968   78.8823   75.1224   82.0750   83.2513   38.0848   91.0536
   56.6621   47.3031   83.1189   95.1907   99.5374   63.4579   20.6431
   81.1319   82.8802   92.2338    7.6273   64.9751   36.3229   33.8604


ans =

   94.4029   80.4069   93.2303
   50.7602   46.3295   56.8354
   47.3031   95.1907   63.4579
```

Problem 2:

opposite order

```
clear;clc;
% give random matrix
N = 100*rand(1,10)
% flip order
flip_it(N)
```

```
N =

  # 1 # 7

   57.4126   48.6932   26.2219   57.9593   87.8328    6.0950   44.0876

  # 8 # 10

    8.4258   56.3238   53.9311


ans =

  # 1 # 7

   53.9311   56.3238    8.4258   44.0876    6.0950   87.8328   57.9593

  # 8 # 10

   26.2219   48.6932   57.4126
```

Problem 3:

top right corner of N

```
clear;clc;
% random matrix N's size and reduced matrix's size
x=fix(rand()*10);
y=fix(rand()*15);
n = min(x,y)-1;
% random matrix N
N = 100*rand(x,y)
% top right matrix
top_right(N,n)
```

```
N =

   58.7362   95.2528    0.6226
   45.8974   29.8201   37.4346
   86.0982   15.8406   90.1496
   66.0836   36.1297   31.8345
   35.3879   74.1629   59.7083
   34.7186   70.5900   29.7795
   25.3718   70.0892   12.5014
```

```
ans =

   95.2528    0.6226
   29.8201   37.4346
   15.8406   90.1496
   36.1297   31.8345
   74.1629   59.7083
   70.5900   29.7795
   70.0892   12.5014
```

Problem 4:

adds together the elements in the first and last rows and columns random matrix N's size

```
clear;clc;
x=fix(rand()*10+2);
y=fix(rand()*15+2);
% random matrix N
N = 100*rand(x,y)
% give out peri_sum
peri_sum(N)
```

```
N =

  # 1 # 7

   98.1176   31.7805   16.6890   71.7470   85.9717    6.6677   20.8258
   86.1990   98.4448   90.3098   13.3432   67.7725   54.1518   60.8161
    8.3821   54.8251   10.5124   44.5789   80.5838   28.1660   32.6176
   33.7712   74.9251   74.5093   50.8787   53.1243   48.0900   88.0847
   23.6129   84.1852   72.9372   53.0490   95.5896   68.4864   13.3395

  # 8 # 14

   10.2408    8.9569   71.2711   30.5053   61.9387   73.5733    5.2211
   95.9117   45.4425   47.2598   78.9811   61.5329   41.1308   57.1186
   15.2902   66.8896   70.8588   23.6387   12.2624   82.8982   74.7670
   15.2538   83.1302   95.8059   23.4303   12.3794   93.5114   32.0244
   15.5553   79.0235   50.5776   46.4699   28.4459   39.9067   49.2934


ans =

   1.6062e+03
```

Problem 5:

```
clear;clc
% the sum of n_th above terms
```

```
sum = 0;
n = 0;
x=pi/2;
% function of term in series
f= @(m,x) (-1)^((m-1)/2)*x^m/factorial(m);
% the error between sin(x) and sum
err=sin(x);
while(1)
    % precision judgement
    if(abs(err)<0.01)
        break;
    end
    % give out each term's value
    term = f(2*n+1,x);
    n = n+1;
    % add rach term to sum
    sum = sum+term;
    err = (sin(x)-sum)/sin(x);
end
fprintf("%d terms are needed",n)
```

```
3 terms are needed
```

Problem 6:

the height after the 8th bounce?

```
clear;clc
hInitial=2;
hFinal=(0.85^2)^8;
fprintf("The height after the 8^(th)bounce is %.2f meters",hFinal)
```

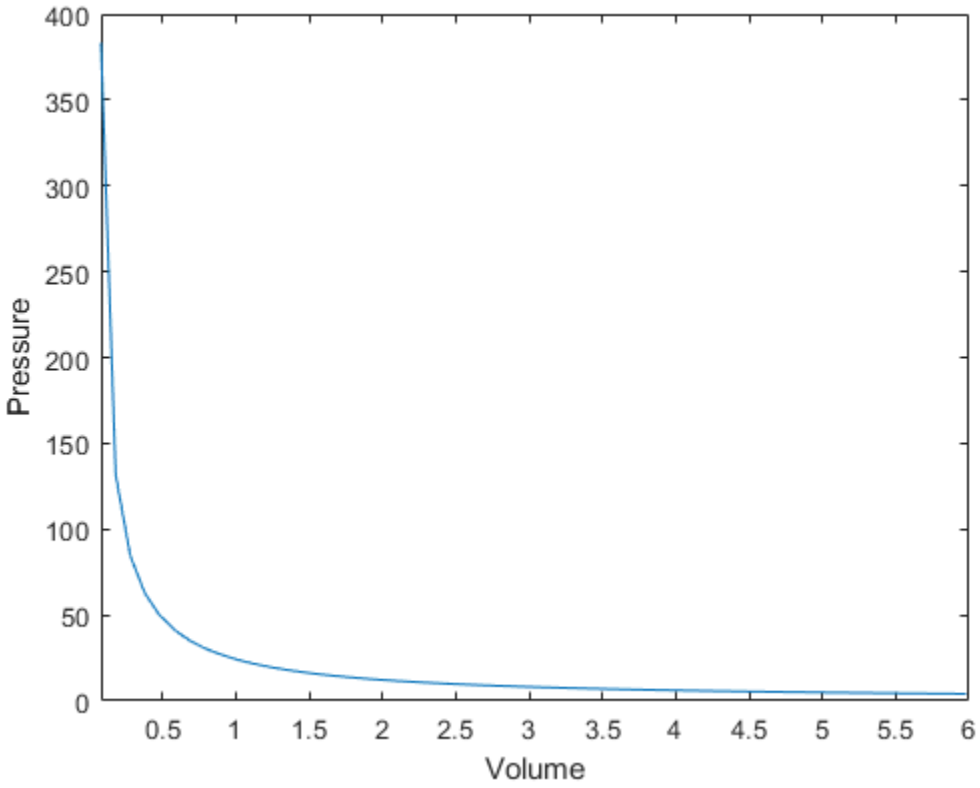
```
The height after the 8^(th)bounce is 0.07 meters
```

Problem 7:

pressure vs. volume

```
clear;clc
% parameter
V = 0.08:0.1:6;
n=1;
T=300;
R=0.08206;
a=1.39;
b=0.039;
% funciton of P, as V is independent variable
P = n*R*T./(V-n*b)-n^2*a./(V.^2);
% draw the picture
plot(V,P)
axis([0.08,6,0,400])
% xlabel
xlabel("Volume");
```

```
% ylabel
ylabel("Pressure");
```



Problem 8:

```
clear;clc;
% random a matrix x
x = 10*rand(ceil(10*rand)+2,1);
mysum=0;
% adding each term to get sum
for li=1:size(x)
    mysum = mysum+x(li);
end
% answer judgement
if mysum == sum(x)
    disp('Congratulations!!, you did it right')
    load handel; sound(y,Fs)
else
    fprintf('Sorry, %.2f != %.2f. Please try again.\n',mysum,sum(x))
end
% repeat but use 'while' loop
mysum=0;
lj=1;
[r,c]= size(x);
while(lj<=r)
    mysum = mysum+x(lj);
```

```
    lj=lj+1;
end
% answer judgement
if mysum == sum(x)
    disp('Congratulations!!, you did it right')
    load handel; sound(y,Fs)
else
    fprintf('Sorry, %.2f != %.2f. Please try again.\n',mysum,sum(x))
end
```

```
Congratulations!!, you did it right
Congratulations!!, you did it right
```

Attachment of function

```
clear;clc;

% Problem 1:
function r = even_index(M)
% get size of M
[r,c] = size(M);
A = [];
% get the even row and column element when both of row and column >1
if (r>1)
    if(c>1)
        for li=2:2:r
            % add new element in to new row B
            B = [];
            for lj=2:2:c
                B=[B,M(li,lj)];
            end
            % add new row into new matrix A
            A =[A;B];
        end
    else
        % row > 1 but column = 1
        % one time traverse
        for li=2:2:r
            A = [A;M(li,lj)];
        end
    end
else
    % row = 1 but column > 1
    % one time traverse
    if(c>1)
        for lj=2:2:c
            A = [A;M(li,lj)];
        end
    else
        % row = 1 and column = 1
        % no answer
        fprintf("sorry, no answer")
    end
```

```
end
r=A;
end

% Problem 2:
function f=flip_it(N)
% new matrix A to keep flip ordered elements
A = [];
% get the size of N
[r,c]=size(N);
for li=c:-1:1
    %traverse the matrix N and add element into A in flipped order
    A = [A,N(1,li)];
end
f = A;
end

% Problem 3:
function f = top_right(N,n)
% get the size of N
[r,c]=size(N);
% new matrix A to keep flip ordered elements
A=[];
for li = 1:r
    % add new element in to new row B
    B=[];
    for lj = c-n+1:c
        B=[B N(li,lj)];
    end
    % add new row into new matrix A
    A = [A;B];
end
f = A;
end

% Problem 4:
function f = peri_sum(N)
% get the size of N
[r,c] = size(N);
sum = 0;
% sum of first raw
for lj = 1:c
    sum = sum + N(1,lj);
end
% sum of last raw
for lj = 1:c
    sum = sum + N(r,lj);
end
% if raw > 2
% sum of first column except the firt one and last one
% "if" judgement is not necessary,if 2<r-1 no error in matlab
% li will be null
if(r>2)
    for li = 2:r-1
```

```
            sum = sum + N(li,1);
        end
        for li = 2:r-1
            sum = sum + N(li,c);
        end
    end
    f=sum;
    end
```

Published with MATLAB® R2020b