

---

```

clc
clear
%The above lines ensure a clean slate for creating variables and a new
%script

%%Problem 1
%Problem 1A
a=4*(pi)^2

%Problem 1B
A=[1 2 3]
B=[4 5 6]
C=[7 8 9]

AA=sum(A)
BB=sum(B)
CC=sum(C)
%The above 3 lines take the sum of the matrices stored in the defined
%variables: A,B,C and redefines them to the respective variables:
AA,BB,CC
Product=AA*BB*CC

%Problem 1C
e=exp(1)
%the above line defines the variable e which is used in the line below
avrg=mean(2.1*[5; 17; 18]+[e; 5^(.5);1.27^(2.2)])

%Problem 1D
A1=[1 0 1 0; 0 0 1 0; 1 1 0 1; 0 1 0 0]
B2=[A1 A1 A1 A1 A1 A1]
C3=[A1; A1; A1; A1; A1; A1]
D=C3*B2
%The above 4 lines are all variables that hold contents provided to us
in
%the lab 8 pdf file
E=nnz(D)
%The line above takes the contents stored in the variable D and
calculates
%how many nonzero values exist then redefines that calculation in the
%variable E

a =

    39.4784

A =

     1     2     3

```

---

---

*B* =

4      5      6

*C* =

7      8      9

*AA* =

6

*BB* =

15

*CC* =

24

*Product* =

2160

*e* =

2.7183

*avrg* =

30.2154

*A1* =

1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0

*B2* =

*Columns 1 through 13*

	1	0	1	0	1	0	1	0	1	0	1
0	1										

---

	0	0	1	0	0	0	1	0	0	0	1
0	0										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	0	1	0	0	0	1	0	0	0	1	0
0	0										

*Columns 14 through 24*

	0	1	0	1	0	1	0	1	0	1	0
	0	1	0	0	0	1	0	0	0	1	0
	1	0	1	1	1	0	1	1	1	0	1
	1	0	0	0	1	0	0	0	1	0	0

$C3 =$

1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	0

$D =$

*Columns 1 through 13*

	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										

---

	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										
	2	1	1	1	2	1	1	1	2	1	1
1	2										
	1	1	0	1	1	1	0	1	1	1	0
1	1										
	1	1	2	0	1	1	2	0	1	1	2
0	1										
	0	0	1	0	0	0	1	0	0	0	1
0	0										

*Columns 14 through 24*

	1	1	1	2	1	1	1	2	1	1	1
	1	0	1	1	1	0	1	1	1	0	1
	1	2	0	1	1	2	0	1	1	2	0
	0	1	0	0	0	1	0	0	0	1	0
	1	1	1	2	1	1	1	2	1	1	1
	1	0	1	1	1	0	1	1	1	0	1
	1	2	0	1	1	2	0	1	1	2	0
	0	1	0	0	0	1	0	0	0	1	0
	1	1	1	2	1	1	1	2	1	1	1

---

---

1	0	1	1	1	0	1	1	1	0	1
1	2	0	1	1	2	0	1	1	2	0
0	1	0	0	0	1	0	0	0	1	0
1	1	1	2	1	1	1	2	1	1	1
1	0	1	1	1	0	1	1	1	0	1
1	2	0	1	1	2	0	1	1	2	0
0	1	0	0	0	1	0	0	0	1	0
1	1	1	2	1	1	1	2	1	1	1
1	0	1	1	1	0	1	1	1	0	1
1	2	0	1	1	2	0	1	1	2	0
0	1	0	0	0	1	0	0	0	1	0
1	1	1	2	1	1	1	2	1	1	1
1	0	1	1	1	0	1	1	1	0	1
1	2	0	1	1	2	0	1	1	2	0
0	1	0	0	0	1	0	0	0	1	0
1	1	1	2	1	1	1	2	1	1	1
1	0	1	1	1	0	1	1	1	0	1
1	2	0	1	1	2	0	1	1	2	0
0	1	0	0	0	1	0	0	0	1	0

$E =$

396

```
%Problem 2
array1 = randi(5,1,100)
%in the parenthesis above the 5 indicates the max value of the
  integers
%being created. the default minimum integer value is 1.
%the 1 indicated the amount of rows being created in the array
%the 100 indicated the amount of collumns being created in the array
histogram(array1,10)
%the above line creates a histogram using the values generated in the
  array
%the number 10 indicates we want 10 bins
avgarr1=mean(array1)
%the above line calculates the mean value of all the values generated
  in
%the array.
```

array1 =

*Columns 1 through 13*

3	3	5	2	2	1	5	4	3	4	3
4	3									

*Columns 14 through 26*

4	3	5	2	1	1	1	3	3	2	4
4	4									

*Columns 27 through 39*

---

	5	5	1	1	4	1	3	3	5	3	2
4	4										

*Columns 40 through 52*

	3	2	1	3	2	1	4	2	3	4	2
4	2										

*Columns 53 through 65*

	4	4	3	1	2	3	2	1	5	3	5
2	4										

*Columns 66 through 78*

	2	5	4	2	2	4	5	2	4	3	5
4	1										

*Columns 79 through 91*

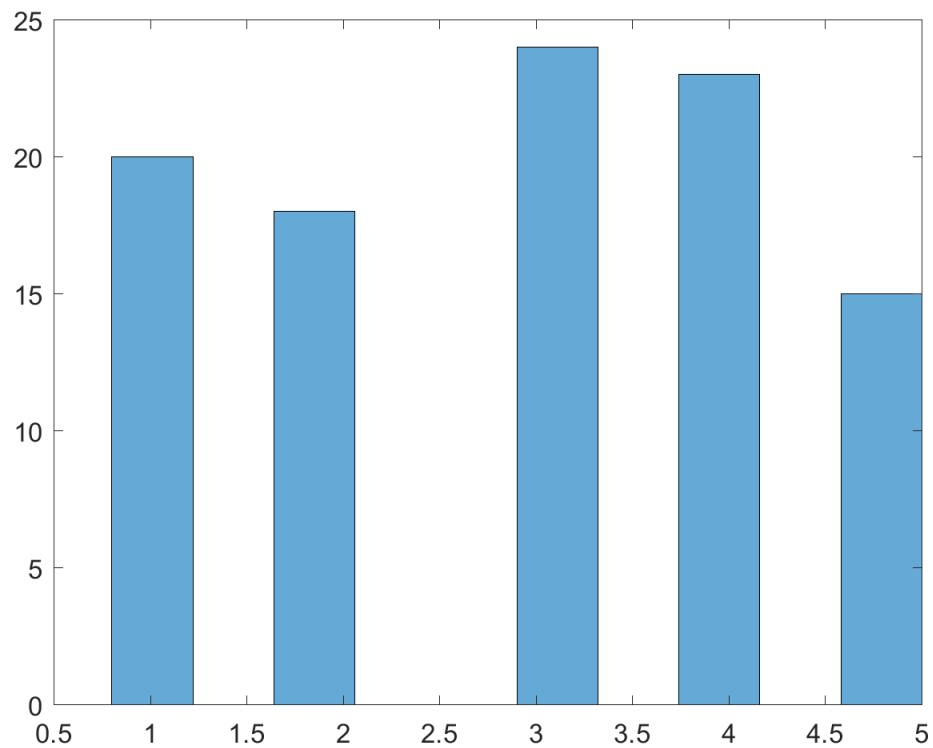
	5	5	3	5	3	1	1	3	4	5	4
2	3										

*Columns 92 through 100*

1	1	1	4	3	1	3	1	1			
---	---	---	---	---	---	---	---	---	--	--	--

*avgarr1 =*

*2.9500*



```
%Problem 3
filename='lab6data.xls'
%the line above sets the variable called filename to reference the
excel
%file saves as lab6data
sheet=1
%the line above indicates that within the excel file we want to
reference
%the first sheet

xlRange1='A98:A169'
xlRange2='B98:B167'
xlRange3='C98:C163'
%the above lines indicate the range of cells from the excel document
we want to
%reference for each indicated 'run'. It saves the parameters for each
run in
%the respected variables listed above represented run 1, 2 and 3

accA= xlsread(filename,sheet,xlRange1);
MaxaccA= max(accA)
%The above 2 lines utilize the variables we specified above with the
first range variable and then uses
%the xlsread function to pull the data from the excel document into
matlab.
```

---

```

%It indicates to save this data in the variable accA, representing the
    first run, and then proceeds to
%analyze that data to find the maximum acceleration value that exists.
    It
%takes this maximum accereration value and stores it in the variable
%MaxaccA

accB= xlsread(filename,sheet,xlRange2);
MaxaccB= max(accB)
%The above 2 lines utilize the variables we specified above with the
    second range variable and then uses
%the xlsread function to pull the data from the excel document into
    matlab.
%It indicates to save this data in the variable accB, representing the
    second run, and then proceeds to
%analyze that data to find the maximum acceleration value that exists.
    It
%takes this maximum accereration value and stores it in the variable
%MaxaccB

accC= xlsread(filename,sheet,xlRange3);
MaxaccC= max(accC)
%The above 2 lines utilize the variables we specified above with the
    third range variable and then uses
%the xlsread function to pull the data from the excel document into
    matlab.
%It indicates to save this data in the variable accC, representing the
    third run, and then proceeds to
%analyze that data to find the maximum acceleration value that exists.
    It
%takes this maximum accereration value and stores it in the variable
%MaxaccC

filename =

    'lab6data.xls'

sheet =

    1

xlRange1 =

    'A98:A169'

xlRange2 =

    'B98:B167'

```

---



---

```
xlRange3 =  
    'C98:C163'
```

```
MaxaccA =  
    42.4000
```

```
MaxaccB =  
    15.3000
```

```
MaxaccC =  
    17
```

```
Published with MATLAB® R2017a
```