# **Introduction to Programming with MATLAB**

# **Vanderbilt University**

# **SLOTIONS**

## **Week-2**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MATLAB as a Calculator \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  x=1000+(1000\*10/100); |
| debt=x+(x\*10/100) |
|  |
|  |
|  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lesson 1 Wrap-up \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| y=0.1/9.58; |
| hundred=y\*3600 |
| x=42.195/((121\*60)+39); |
| marathon=x\*3600 |

## **Week-3**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Colon Operator Practice \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| odds=1:2:100 |  |
| evens=100:-2:1 |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Matrix Indexing Practice \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| A = [1:5; 6:10; 11:15; 16:20]; |  |
| v=A(:,2) |  |
| A(end,:)=0 |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Matrix Arithmetic \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| A = [1:5; 6:10; 11:15; 16:20]; |  |
| x=ones(1,size(A,1)) |  |
| y=ones(size(A,2),1) |  |
| result=x\*A\*y |  |

## **Week-4**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* A Simple Function \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| function area=tri\_area(b,h) |  |
| area=0.5\*b\*h |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Corner Case \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function [top\_left top\_right bottom\_left bottom\_right] = corners(A) |  |
| top\_left=A(1,1) |  |
| top\_right=A(1,end) |  |
| bottom\_left=A(end,1) |  |
| bottom\_right=A(end,end) |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Taxi Fare \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function fare=taxi\_fare(d,t) |  |
| fare=5+(2\*(ceil(d)-1))+(0.25\*ceil(t)) |  |

## **Week-5**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Minimum and Maximum \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| function [mmr mmm] = minimax(M) |  |
| mmr=max(M')-min(M') |  |
| mmm=max(max(M))-min(min(M)) |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Matrix Construction \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function M=trio(n,m) |  |
| M=[ones(n,m);2\*ones(n,m);3\*ones(n,m)] |  |

## **Week-6**

|  |  |
| --- | --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Practice if-statements \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |
| function out = picker(condition,in1,in2) |  | |
| if condition>0 |  | |
| out=in1 |  | |
| else |  | |
| out=in2 |  | |
| end |  | |
|  |  | |
|  |  | |
|  |  | |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* More Practice \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  | |
| function admit = eligible(v,q) |  | |
| avg=(v+q)/2; |  | |
| if avg>=92&&v>88&&q>88 |  | |
| admit=1>0 |  | |
| else |  | |
| admit=0>1 |  | |
| end |  | |
|  |  | |
|  |  | |
|  |  | |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Variable Number of Input Arguments \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  | |
| function too\_young = under\_age(age,limit) |  | |
| if nargin==1 |  | |
| limit=21; |  | |
| end |  | |
| if age<limit |  | |
| too\_young=1>0 |  | |
| else |  | |
| too\_young=1<0 |  | |
| end |  | |
|  |  | |
|  |  | |
|  |  | |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lesson 5 Wrap-up \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  | |
| function valid = valid\_date(y,m,d) |  | |
| if ~isscalar(y)||~isscalar(m)||~isscalar(d) |  | |
| valid=1<0 |  | |
| else |  | |
| if m>=1&&m<=12 |  | |
| if m==1||m==3||m==5||m==7||m==8||m==10||m==12 |  | |
| if d>=1&&d<=31 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| elseif m==4||m==6||m==9||m==11 |  | |
| if d>=1&&d<=30 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| else |  | |
| if rem(y,100)==0 |  | |
| if rem(y,400)==0 |  | |
| if d>=1&&d<=29 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| else |  | |
| if d>=1&&d<=28 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| end |  | |
| elseif rem(y,4)==0 |  | |
| if d>=1&&d<=29 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| else |  | |
| if d>=1&&d<=28 |  | |
| valid=1>0 |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| end |  | |
| end |  | |
| else |  | |
| valid=1<0 |  | |
| end |  | |
| end |  | |

## **Week-7**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Practice for-loops \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| function summa= halfsum(A) |  |
| total=0; |  |
| for i=1:size(A,1) |  |
| for j=size(A,2):-1:i |  |
| total=total+A(i,j) |  |
| end |  |
| end |  |
| summa=total; |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Practice while-loops \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function k = next\_prime(n) |  |
| prime=n+1; |  |
| while isprime(prime)~=1 |  |
| prime=prime+1; |  |
| end |  |
| k=prime |  |
|  |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Logical Arrays Practice \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function numfreeze = freezing(temp) |  |
| numfreeze=sum(temp<32) |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Lesson 6 Wrap-up \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function [summa index] = max\_sum(v,n) |  |
| maxv=0; |  |
| ind=-1; |  |
| if n>size(v,2) |  |
| summa=0 |  |
| index=-1 |  |
| else |  |
| for i=1:n |  |
| maxv=sum(v(1:n)); |  |
| ind=1; |  |
| end |  |
| for j=2:(size(v,2)-n+1) |  |
| total=0; |  |
| total=sum(v(j:(j+n-1))); |  |
| if total>maxv |  |
| maxv=total; |  |
| ind=j; |  |
| else |  |
| continue; |  |
| end |  |
| end |  |
| end |  |
| summa=maxv |  |
| index=ind |  |

## **Week-8**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Simple Encryption \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| function encoded = caesar(str,n) |  |
| strascii=double(str); |  |
| temp=strascii+n; |  |
| for i=1:numel(temp) |  |
| while (temp(i)>126||temp(i)<32) |  |
| if temp(i)>126 |  |
| diff=temp(i)-126; |  |
| temp(i)=31+diff; |  |
| else |  |
| diff=32-temp(i); |  |
| temp(i)=127-diff; |  |
| end |  |
| end |  |
| cipher(i)=temp(i); |  |
| end |  |
| encoded=char(cipher); |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparse Matrix \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function matrix = sparse2matrix(cellvec) |  |
| sizecell=numel(cellvec); |  |
| default = cellvec{1,2}\*ones(cellvec{1,1}(1,:)); |  |
| for i=3:sizecell |  |
| row=cellvec{1,i}(1); |  |
| col=cellvec{1,i}(2); |  |
| val=cellvec{1,i}(3); |  |
| default(row,col)=val; |  |
| end |  |
| matrix=default; |  |

## **Week-9**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Excel File I/O \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| function distance = get\_distance(city1,city2) |  |
| [num,text]=xlsread('Distances.xlsx'); |  |
| row=-1; |  |
| column=-1; |  |
| rowcity=text(1,:); |  |
| columncity=text(:,1); |  |
| for i=2:numel(rowcity) |  |
| if strcmpi(city1,rowcity(i)) |  |
| row=i; |  |
| else |  |
| continue; |  |
| end |  |
| end |  |
| for i=2:numel(columncity) |  |
| if strcmpi(city2,columncity(i)) |  |
| column=i; |  |
| else |  |
| continue; |  |
| end |  |
| end |  |
| if row==-1||column==-1 |  |
| distance=-1; |  |
| else |  |
| distance=num(row-1,column-1); |  |
| end |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Text File I/O \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function charnum = char\_counter(fname,ch) |  |
| fid=fopen(fname,'rt'); |  |
| if fid<0 |  |
| charnum=-1; |  |
| return; |  |
| end |  |
| oneline = fgets(fid); |  |
| str=[oneline]; |  |
| while ischar(oneline) |  |
| oneline=fgets(fid); |  |
| str=[str oneline]; |  |
| end |  |
| count = 0; |  |
| if (ischar(ch)) |  |
| for i=1:numel(str) |  |
| if ch==str(i) |  |
| count=count+1; |  |
| else |  |
| continue; |  |
| end |  |
| end |  |
| charnum=count; |  |
| else |  |
| charnum = -1; |  |
| end |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Saddle Points \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function indices = saddle(M) |  |
| indices=[]; |  |
| [row,col]=size(M); |  |
| for i=1:col |  |
| for j=1:row |  |
| minM=min(M(:,i)); |  |
| maxM=max(M(j,:)); |  |
| if M(j,i)==minM && minM==maxM |  |
| indices=[indices;j i]; |  |
| end |  |
| end |  |
| end |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Image blur \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function output = blur(img,w) |  |
| [r c]=size(img); |  |
| A=(ones((r+2\*w),(c+2\*w)))\*300; |  |
| A(w+1:w+r,w+1:w+c)=img; |  |
| S=(A<300).\*A; |  |
| for i=w+1:w+r |  |
| for j=w+1:w+c |  |
| dig=0; |  |
| output((i-w),(j-w))=(sum(sum(S(i-w:i+w,j-w:j+w)))); |  |
| dig=sum(sum(A(i-w:i+w,j-w:j+w)<270)); |  |
| output((i-w),(j-w))=output((i-w),(j-w))/dig; |  |
| end |  |
| end |  |
| output=uint8(output); |  |
|  |  |
|  |  |
|  |  |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Echo Generator \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |
| function output = echo\_gen(input, fs, delay, amp) |  |
| samples=round(fs\*delay); |  |
| if samples>0 |  |
| x=zeros(samples,1); |  |
| else |  |
| x=[]; |  |
| end |  |
| y=amp\*input; |  |
| echo=[x ; y]; |  |
| n=[input ; x]; |  |
| w = n+echo; |  |
| if max(w)>1 |  |
| output=w/max(w); |  |
| else |  |
| output=w; |  |
| end |  |