MATLAB PROJECT

GENN004: Computers For Engineers

PROJECT: SUDOKU Solver

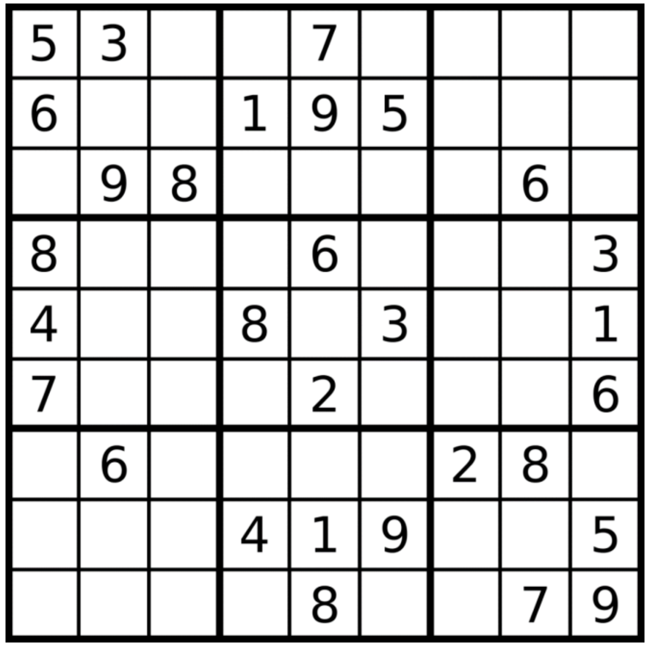
PROJECT: SUBMITTED TO DR. IBRAHIM YOUSSEF

1) Islam Amr Abdelmoneim Mohamed (1200487) (islam.mohamed02@eng-st.cu.edu.eg)
2) Alhussein Gamal Hussein Ali (1200399) (alhussein.ali02@eng-st.cu.edu.eg)
3) Mostafa Usama Mahmoud Ahmed Elkholy (1200495) (mostafa.ahed02@eng-st.cu.edu.eg)
4) Mohamed Gamal Hussein Ali (1200435) (mohamed.ali023@eng-st.cu.edu.eg)
5) Mohamed Ahmed Mohamed Kamel Elewa (1200433) (mohamed.kamel02@eng-st.cu.edu.eg)

SUDOKU

What is sudoku?

- Sudoku is a logic-based, number-placement puzzle
- You have to fill a 9x9 grid with numbers from 1-9
- Each column, row and 3x3 grid should have all numbers from 1-9



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

PROGRAM

The sudoku program has these features

- Solve the puzzle
- Display the sudoku grid
- Use maximum of 3 hints
- Get possible values that could be assigned to a cell.
- Save the progress
- Reset to the original board
- Quit the game

GUIDE

How can the user use the program?

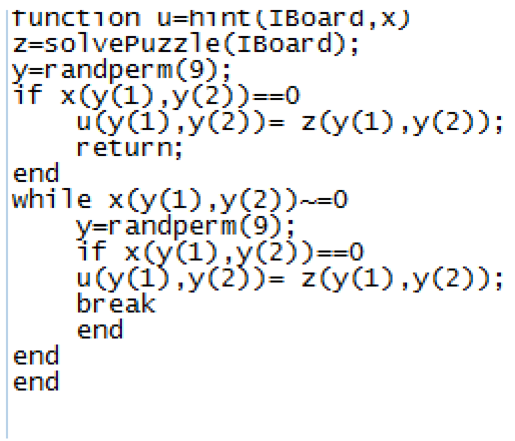
- Play (p)
  1) user enters the number they want to add
  2) user enters a 2-D array for the position of the number
- Solve (s)
  1) The program solves the puzzle by identifying the missing numbers
  2) Displays the grid with the solved puzzle

GUIDE

- Hint (h)
  1) Random correct number is generated
  2) Placed in the grid
  3) Can only be used three times
- Reset (r)

  Resets the grid to original board
- Quit (q)

  Quits the game and saves the progress in another sheet in the excel so that the user has the option to resume on his previous progress.



```
function u=hint(IBoard,x)
z=solvePuzzle(IBoard);
y=randperm(9);
if x(y(1),y(2))==0
    u(y(1),y(2))= z(y(1),y(2));
    return;
end
while x(y(1),y(2))~=0
    y=randperm(9);
    if x(y(1),y(2))==0
    u(y(1),y(2))= z(y(1),y(2));
    break
    end
end
end
```

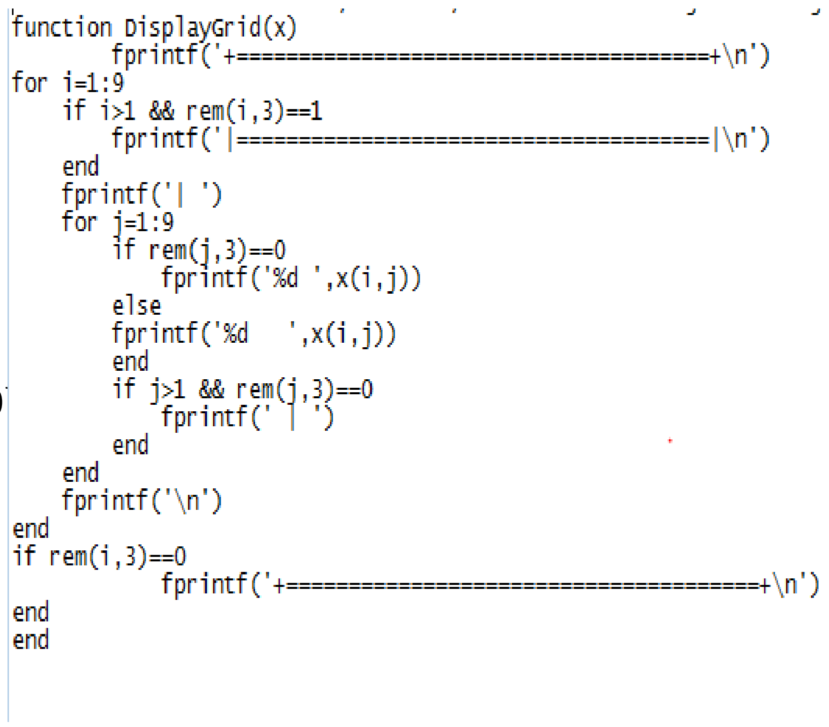
MAIN FUNCTIONS

These are the main functions in the program

- DisplayGrid

It incorporates for loops that fill, for every three rows, the nine columns with zeros (the initial value assigned to empty cells). It also displays the borderlines between each three rows using quotations executed thro

A fprintf funtion.

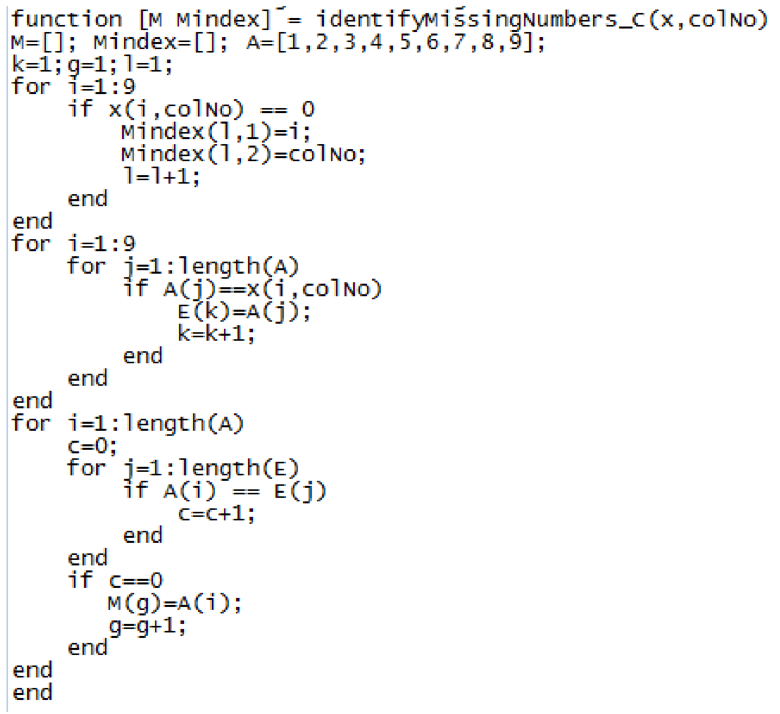


```
function DisplayGrid(x)
        fprintf('+======================================+\n')
for i=1:9
    if i>1 && rem(i,3)==1
        fprintf('|======================================|\n')
    end
    fprintf('| ')
    for j=1:9
        if rem(j,3)==0
            fprintf('%d ',x(i,j))
        else
        fprintf('%d   ',x(i,j))
        end
        if j>1 && rem(j,3)==0
            fprintf(' | ')
        end
    end
    fprintf('\n')
end
if rem(i,3)==0
            fprintf('+======================================+\n')
end
end
```

▪ IdentifyingMissingNumbers_R/C

Through a loop, those two functions identify the missing numbers in each row and column. For 2 nested loops i and j, a function CompareAndEliminate is used to compare these two arrays containing the possible values for each row and column, returning a new array that contains common values between two arrays. The returned array, the common vector, is the possible values

that could be assigned to cell(i,j).



```
function [M Mindex] = identifyMissingNumbers_C(x,colNo)
M=[]; Mindex=[]; A=[1,2,3,4,5,6,7,8,9];
k=1;g=1;l=1;
for i=1:9
    if x(i,colNo) == 0
        Mindex(l,1)=i;
        Mindex(l,2)=colNo;
        l=l+1;
    end
end
for i=1:9
    for j=1:length(A)
        if A(j)==x(i,colNo)
            E(k)=A(j);
            k=k+1;
        end
    end
end
for i=1:length(A)
    c=0;
    for j=1:length(E)
        if A(i) == E(j)
            c=c+1;
        end
    end
    if c==0
       M(g)=A(i);
       g=g+1;
    end
end
end
```

MAIN FUNCTIONS

- IdentifyingMissingIndices

At the beginning of each loop, this function identifies the missing indices in the grid.

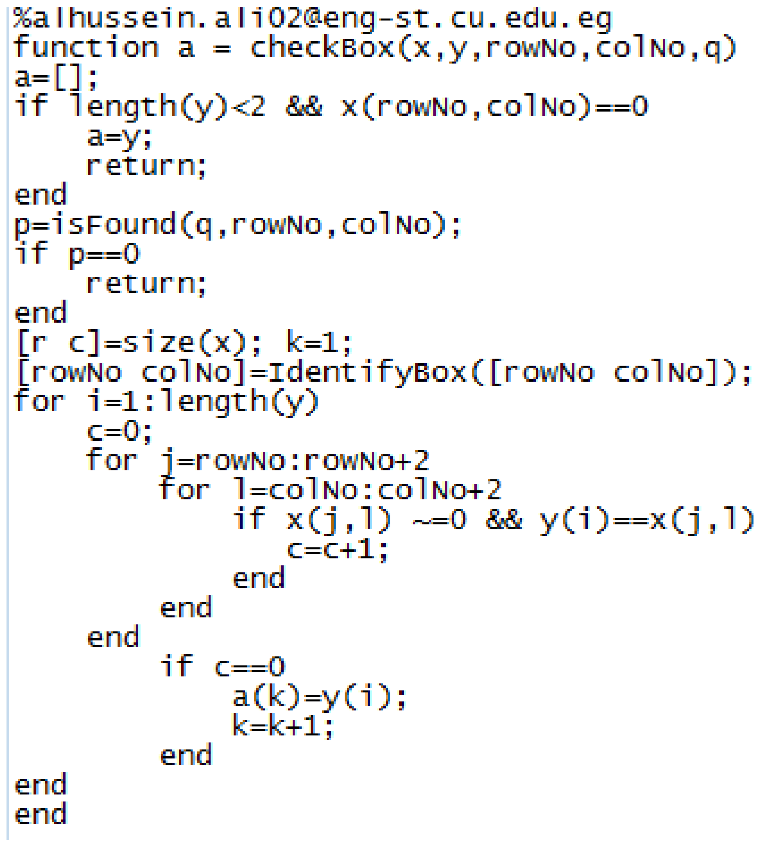
It is constantly updated with every loop run of the previous IdentifyMissingNumbers_R/C functions.

MAIN FUNCTIONS

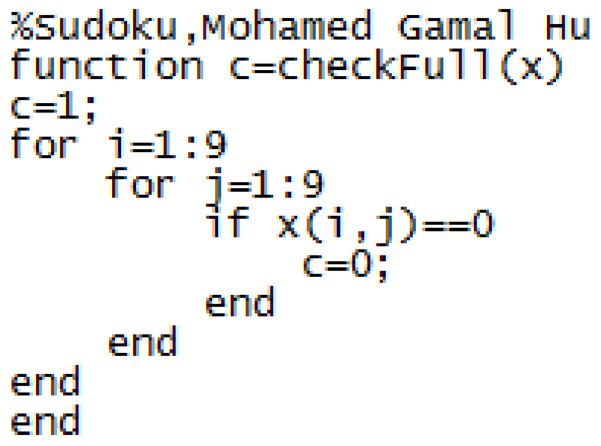
- CheckBox (x,y,i,j,q)

This is probably the most important function in our program. It takes as a parameter the possible values that could be assigned to an empty cell. A function called IdentifyBox checks in which box that cell exists. For instance a cell of index (5,9) exists in the box starting at row 4 and column 7 which is box 6 (box in the middle to right) on the grid. Then, the function checks if a number exists in the box and is found in y, then it’s eliminated from the possible values that could be assigned to that index. For easy and intermediate-level puzzles, this technique is sufficient to always find a cell whose array of possible, assignable values contains only one value which is the only value that could be assigned to that cell

SNIPPET OF CHECKBOX FUNCTION

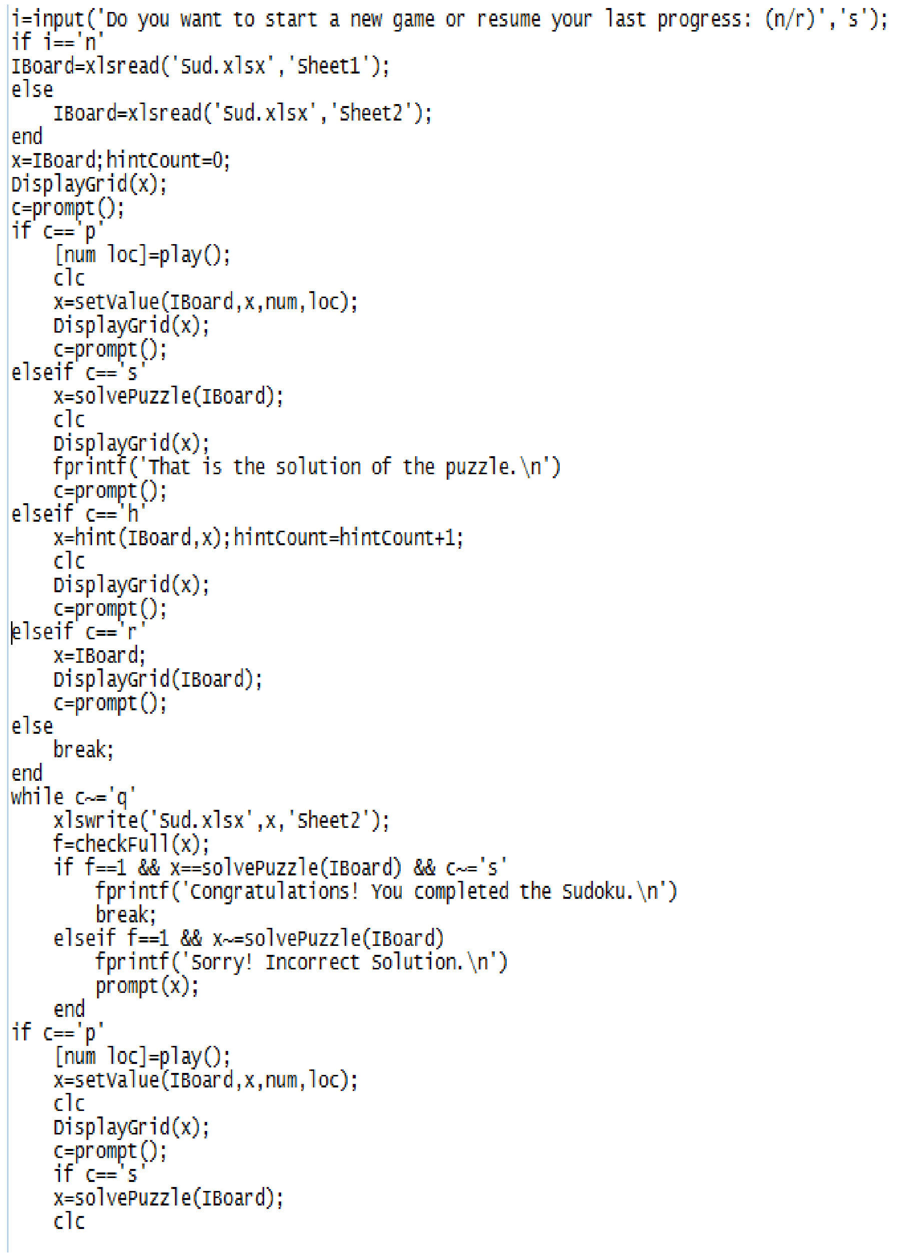


```
%alhussein.ali02@eng-st.cu.edu.eg
function a = checkBox(x,y,rowNo,colNo,q)
a=[];
if length(y)<2 && x(rowNo,colNo)==0
    a=y;
    return;
end
p=isFound(q,rowNo,colNo);
if p==0
    return;
end
[r c]=size(x); k=1;
[rowNo colNo]=IdentifyBox([rowNo colNo]);
for i=1:length(y)
    c=0;
    for j=rowNo:rowNo+2
        for l=colNo:colNo+2
            if x(j,l) ~=0 && y(i)==x(j,l)
               c=c+1;
            end
        end
    end
        if c==0
            a(k)=y(i);
            k=k+1;
        end
end
end
```



```
%Sudoku,Mohamed Gamal Hu
function c=checkFull(x)
c=1;
for i=1:9
    for j=1:9
        if x(i,j)==0
            c=0;
        end
    end
end
end
```

- main



```
i=input('Do you want to start a new game or resume your last progress: (n/r)','s');
if i=='n'
IBoard=xlsread('Sud.xlsx','Sheet1');
else
    IBoard=xlsread('Sud.xlsx','Sheet2');
end
x=IBoard;hintCount=0;
DisplayGrid(x);
c=prompt();
if c=='p'
    [num loc]=play();
    clc
    x=setValue(IBoard,x,num,loc);
    DisplayGrid(x);
    c=prompt();
elseif c=='s'
    x=solvePuzzle(IBoard);
    clc
    DisplayGrid(x);
    fprintf('That is the solution of the puzzle.\n')
    c=prompt();
elseif c=='h'
    x=hint(IBoard,x);hintCount=hintCount+1;
    clc
    DisplayGrid(x);
    c=prompt();
elseif c=='r'
    x=IBoard;
    DisplayGrid(IBoard);
    c=prompt();
else
    break;
end
while c~='q'
    xlswrite('Sud.xlsx',x,'Sheet2');
    f=checkFull(x);
    if f==1 && x==solvePuzzle(IBoard) && c~='s'
        fprintf('Congratulations! You completed the Sudoku.\n')
        break;
    elseif f==1 && x~=solvePuzzle(IBoard)
        fprintf('Sorry! Incorrect Solution.\n')
        prompt(x);
    end
if c=='p'
    [num loc]=play();
    x=setValue(IBoard,x,num,loc);
    clc
    DisplayGrid(x);
    c=prompt();
    if c=='s'
    x=solvePuzzle(IBoard);
    clc
```

solvePuzzle function: function that solves the entire puzzle if the user presses 's.'

This function incorporates the results produced

From the previous checkBox function

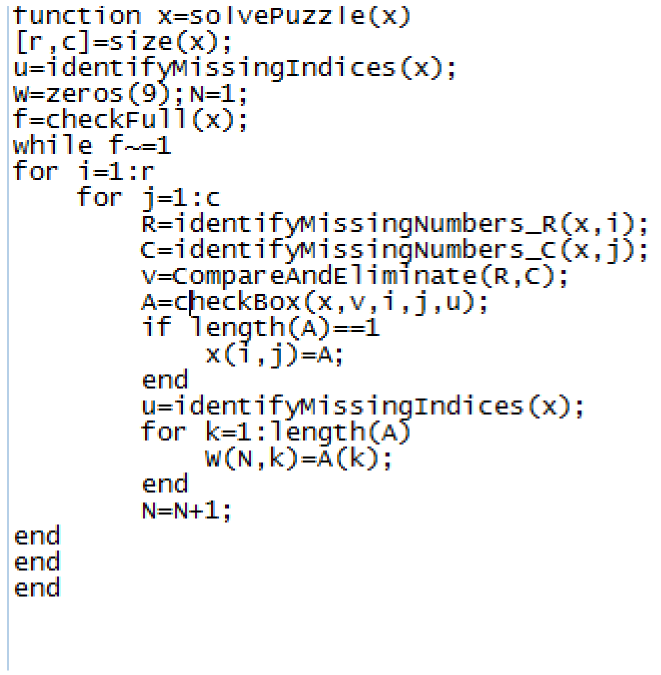
then that value is assigned to that cell.

The loop inside continues as long as the value of checkFull is = 0,

where checkFull is a function that checks whether the array

contains any zero cells or not, where zero defines an empty cell.

As such, the puzzle is solved.



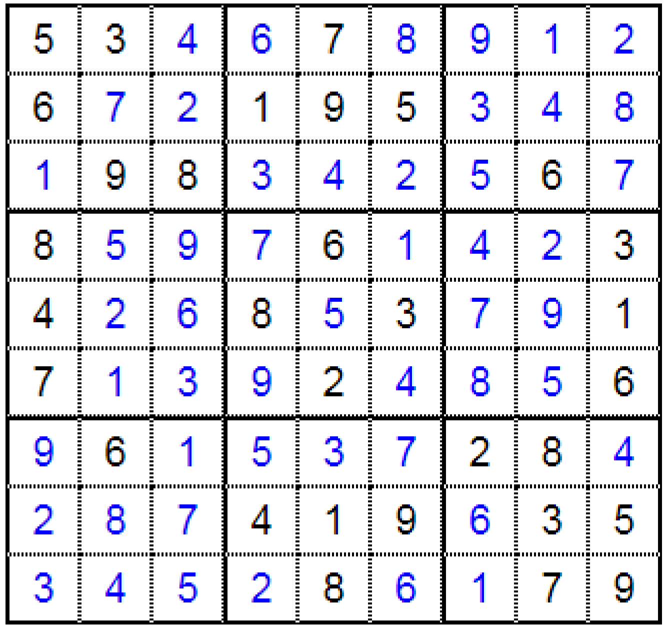
```
function x=solvePuzzle(x)
[r,c]=size(x);
u=identifyMissingIndices(x);
W=zeros(9);N=1;
f=checkFull(x);
while f~=1
for i=1:r
    for j=1:c
        R=identifyMissingNumbers_R(x,i);
        C=identifyMissingNumbers_C(x,j);
        v=CompareAndEliminate(R,C);
        A=checkBox(x,v,i,j,u);
        if length(A)==1
            x(i,j)=A;
        end
        u=identifyMissingIndices(x);
        for k=1:length(A)
            W(N,k)=A(k);
        end
        N=N+1;
end
end
end
```

Sample Run

input

- [5,3,0,0,7,0,0,0,0;
-   6,0,0,1,9,5,0,0,0;
-   0,9,8,0,0,0,0,6,0;
-   8,0,0,0,6,0,0,0,3;
-   4,0,0,8,0,3,0,0,1;
-   7,0,0,0,2,0,0,0,6;
-   0,6,0,0,0,0,2,8,0;
-   0,0,0,4,1,9,0,3,5;
-   0,0,0,0,8,0,0,7,9]

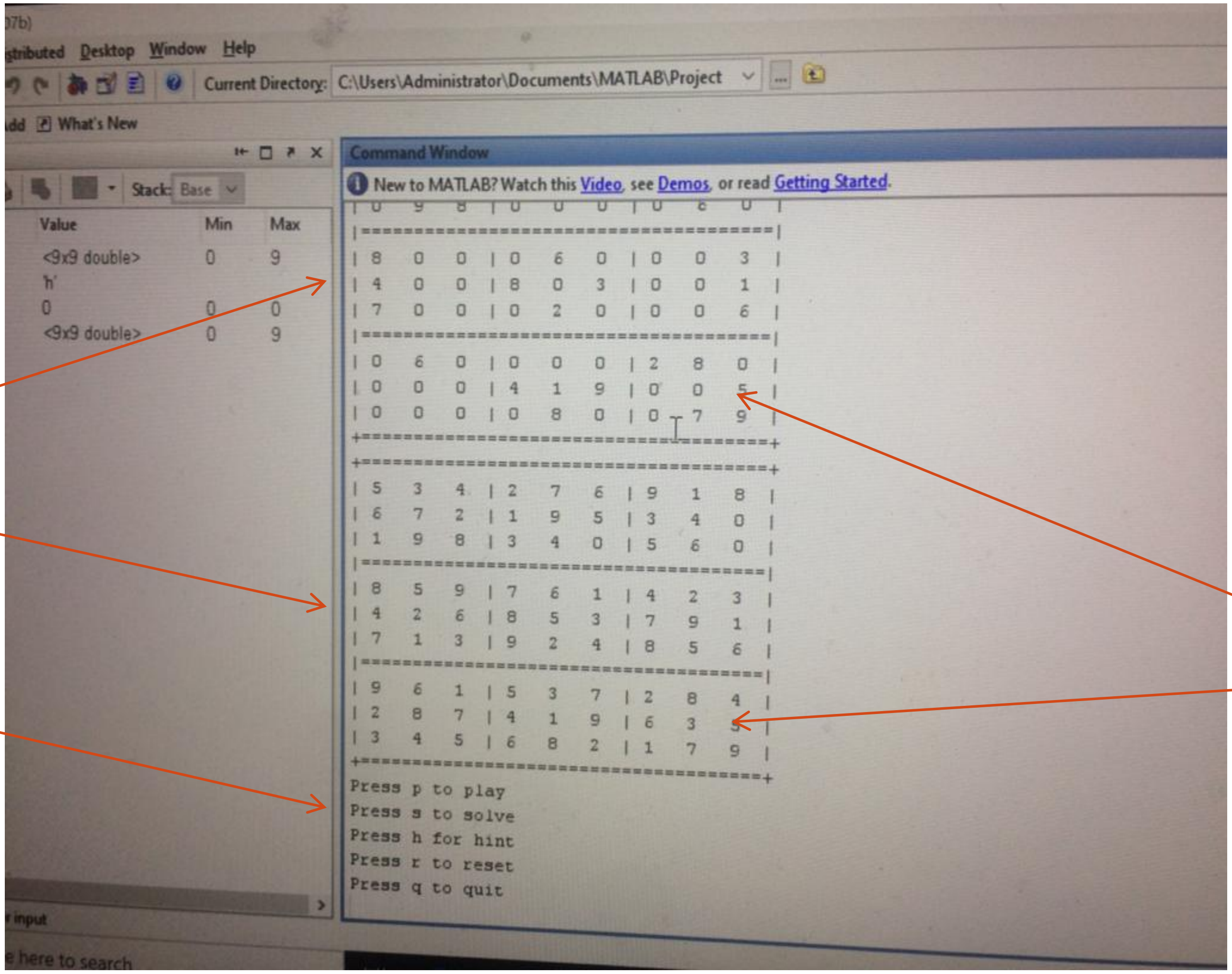
output



5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Solved puzzle (observe 2nd 3x9 grid)

Running Input


07b)
stributed Desktop Window Help
Current Directory: C:\Users\Administrator\Documents\MATLAB\Project
dd What's New
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Value	Min	Max
<9x9 double>	0	9
'h'		
0	0	0
<9x9 double>	0	9

```
| 0   9   8  | 0   0   0  | 0   6   0  |
|======================================|
| 8   0   0  | 0   6   0  | 0   0   3  |
| 4   0   0  | 8   0   3  | 0   0   1  |
| 7   0   0  | 0   2   0  | 0   0   6  |
|======================================|
| 0   6   0  | 0   0   0  | 2   8   0  |
| 0   0   0  | 4   1   9  | 0   0   5  |
| 0   0   0  | 0   8   0  | 0   7   9  |
+======================================+
+======================================+
| 5   3   4  | 2   7   6  | 9   1   8  |
| 6   7   2  | 1   9   5  | 3   4   0  |
| 1   9   8  | 3   4   0  | 5   6   0  |
|======================================|
| 8   5   9  | 7   6   1  | 4   2   3  |
| 4   2   6  | 8   5   3  | 7   9   1  |
| 7   1   3  | 9   2   4  | 8   5   6  |
|======================================|
| 9   6   1  | 5   3   7  | 2   8   4  |
| 2   8   7  | 4   1   9  | 6   3   5  |
| 3   4   5  | 6   8   2  | 1   7   9  |
+======================================+
Press p to play
Press s to solve
Press h for hint
Press r to reset
Press q to quit
```