Faculty of Engineering
MM1CPM

Worksheet 4 – Displaying text and reading form the keyboard

Q1 Embedded computers

The majority of computers you will come across in your professional life won't be desktop PCs with high quality displays, but highly specialized computers embedded in machinery, these are the so called embedded computers. Very often these computers will only save text data to a log file or show output on a very low resolution display. It is therefore very important to be able to accurately process and display text on computers. An example of an embedded computer is shown in figure 1, this is the flight control computer to a Boeing 737. By discussing the subject with the person sitting next to you write down five more examples of an embedded systems in the space below – try to think of unusual examples.

Figure 1: An example of a display on an embed system. It may not have fancy graphics but this display to a modern computer controlling a Boeing 737's flight path.

Q2: The easiest way to control output in MATLAB.

Make a new script called q2.m and define the following variables. x=1, y=2 and z=3. Run the script, it should print out the variables one after each other. Now add a semicolon to the end of each line and rerun the script. This time there should be no output from MATLAB. Placing a semicolon at the end of a line is the easiest way to control output in MATLAB.

Q3: Using the disp command

a) Use the **disp** command to write a script to print the following poem to the screen. Note that you will not be able to display characters which are not in the alphabet with the **disp** command, so just omit these from the poem at the moment.

“Oh this shiny new computer -
There just isn't nothin' cuter.
It knows everything the world ever knew.
And with this great computer I don't need no writin' tutor,
'Cause there ain't a single thing that it can't do.”

b) Use the **disp** command to write a the following text to the screen:

The speed of the airplane is:
500
m/s

Each line of text should be on a new line and the 500 stored in a variable.

c) Make a new script entitled name_age.m. In the script file define a variable name as your name, and a variable age as your age. Then use the **disp** command to print the following to the screen.

My name is:

Faculty of Engineering
MM1CPM

name
My age is:
age

Where name and age are replaced by the content of the variable name and age.

d) Before modern computer with high resolution displays became common place animations and pictures were often made with text. To practice using the **disp** command, make a new script to draw the rocket shown on the right using the '|','^','/', and '\' characters on your keyboard. You will need to use a **disp** command for each line of the image. The pause(x) command will make MATLAB pause x seconds. Now animate the rocket taking off by putting the **pause** command after each disp command. Define the variable x at the top of the script and set x to 0.5. Now make the rocket accelerate by multiplying x by 0.8 every two disp commands.

```
    |
    |
   / \
  /   \
 /     \
/       \
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
^^^^^^^^^
  /\ /\
 /  \/  \
 **   **
**** ****
```

There is a nice animation of a fluid mechanics simulator, which simulates liquid in a box on youtube (https://www.youtube.com/watch?v=QMYfkOtYYlg). What other ASCII art animations can you find on youtube.

Q4: Accurate formatting of text using sprintf
Disp can only offer limited control of text formatting sprintf can offer far more detailed control.

a) Make a new script that uses **sprintf** and **disp** together to print the following line of text to the screen using the %f format specifier:

"The speed of the airplane is 500 m/s"

The 500 should be stored in a variable.

b) Make a new script to print the following text to the screen

"The depth of the submarine is 200m and it's speed is 5km/hour."

The 200 and 5 should be stored in variables. Note that you can't print a ' with the disp command so you will have to obmit it!

c) Use a single **sprintf** command to print the following sentence

"The cost of <object> on the date <day>/<month>/<year> was <price> pounds."

Where <object> is a string variable defined as 'bread'
<day>/<month>/<year> are number variables defined as 12,3 and 2000 respectively.
<price> is a variable containing a number defined as 1.09.

Hint for the <object> you will have to use the '%s' (string), format specifier.

d) In the lecture we learnt how to control the number of decimal points MATLAB prints to the screen. Use the sprintf command along with the %f format specify to find out how many decimal points MATLAB knows pi to. Try multiplying pi by 10,100,1000 and 1e6.

e) Repeat the above question, but change the %f to a '%e' what happens? What is the difference between '%e' and '%f' format specifiers?

f) Define the variable charge_on_an_electron as 1.60217657e-19. Now use **sprintf** and the '%e' format specifier to write out the following sentence.

"The charge on an electron is 1.60e-9 Coulombs to two d.p. or 1.60217657e-19 Coulombs to 8 d.p."

Q5: Special characters using sprintf
a) In question 3a, you got the computer to print out a poem about how clever computers are. However, because **disp** can not display special characters such as apostrophes the poem did not look quite right. Use **disp** and **sprintf** to print the poem out line by line with the correct punctuation. You will find the answer in the correct lecture notes.

b) Using a single **disp** and a single **sprintf** command make MATLAB print the following poem over four lines:

All doggies go to heaven (or so I've been told).
They run and play along the streets of Gold.
Why is heaven such a doggie-delight?
Why, because there's not a single cat in sight!

c) Use a single **disp** and a single **sprintf** to print out the following text:

The computer swallowed grandma.
Yes, honestly its true.
She pressed 'control' and 'enter'
And disappeared from view.

Q6: Strings as arrays
Strings are arrays containing lists of characters. For example 'Hello my name is Rod' is a string. Strings are really just arrays containing characters and you can use all the tricks you learnt for dealing with arrays in lecture 2 to work with strings. This question lets you practice handling strings in MATLAB.

a) In a new script file define the string 'Hello my name is Rod' as a variable called 'a' and use the **disp** command to display it.

b) Using bracket notation, replace the first letter of Rod with a B, and the last letter of with a b. Display this new string.

c) Did you know that spaces, and returns are also characters? Again using bracket notation, replace the spaces in the string with a * character. Display this new string.

d) Using **:** notation, extract each word hello from the string called 'greetings'. Display this new string.

e) Form a new array using the variable greetings, which will contain 'Hello' five times. Display this string. Hint: z=[b b b]

Q7: ASCII numbers

Computers store and transmit text as numbers between 0 and 255. Every letter on the keyboard has a number assigned to it, for example the letter 'D' is assigned the number 68. A full list of the numbers corresponding to letters is given in the lecture notes. This code is called ASCII code or the American Standard Code for Information Interchange. It was originally developed in the 1960s to make transferring information between different computers easy. Almost without exception every computer in the world uses this code to store and transmit information, be it your PC, iPad, or even the computer onboard a cruise missile. It is important to be aware that this code exists because you will come across it later in mechanics. In this question we will practice using it:

Figure 2: NASA Mars rover - all computers you will meet during your professional life will use ASCII code to store and transmit data - even this NASA mars rover.

a) If you type int16('Hello') MATLAB will turn the string 'Hello' into an array of numbers of ASCII code. Try doing this your self.
b) Using the result what is the ASCII code for the letter 'H'. What is the ASCII code for the letter o.
c) Try typing int16('AaAa'), what is the numerical difference between the ASCII code for 'A' and 'a'
d) Try typing int16('BbBb'), what is the numerical difference between the ASCII code for 'B' and 'b'
e) Define the following array in the work space a=[67 111 109 112 117 116 101 114 115 32 97 114 101 32 99 111 111 108 33] – hint you can copy and paste the data from the pdf on moodle.
f) MATLAB provides the char() command to convert arrays of ASCII numbers back into readable text. What information does the above 'a' contain?
g) The following ASCII numbers are sent to a small robot buggy used to draw a company logo on a parking lot. The robot has a paint spray pen which it can turn off and on. Use MATLAB to convert the ASCII code to human readable text to find out, what shape will the buggy draw on the parking lot?

[83 80 82 65 89 79 78 32 70 79 82 87 65 82 68 32 53 48 32 82 73 71 72 84 32 57 48 32 70 79 82 87 65 82 68 32 53 48 32 82 73 71 72 84 32 57 48 32 70 79 82 87 65 82 68 32 53 48 32 82 73 71 72 84 32 57 48 32 70 79 82 87 65 82 68 32 53 48 32 82 73 71 72 84 32 57 48 32 83 80 82 65 89 79 70 70]

h) Did you that even the number keys on your keyboard have ASCII values associated with them just like the letters. Find out what the ASCII codes are for the number string '1001'. Note you must put single quotes around the 1001, so MATLAB treats it as a string variable not a number variable.

Q8: Input from the keyboard
Getting input from the user is often the first thing a program must do, this question let's you practice getting asking the user questions.

a) Using the **input** command and the **disp** command write a program asks the user for two numbers and them prints the sum of the numbers.

b) Write a program that asks the user to enter three numbers, a,b and c. Then give the answer to the roots to the quadratic equation: for $ax^2+bx+c=0$. You can either use MATLAB's built in roots function or the standard quadratic equation.

c) Write a script to ask the user "What's your name?", then print the imputed string in ASCII numbers.

d) Write a script to ask the user and "What's your favorite food ?" and "What's your name?". Then prints the sentence

xxxx's favorite food is yyyy.

Where xxxx and yyyy are replaced by the name of the user and the food.

e) Write a script that asks the user to input a start value, a stop value. Then use linspace to make an array of 100 points from the start value to the stop value. Take the sin of this array and plot the graph of sin between the start and stop point.

f) Write a script to ask the user for the x and y dimensions of a 2D array. Then make a random matrix of this size. Use the surf command to plot this array.

g) If three resistors are placed in parallel the total resistance can be calculate with

$$\frac{1}{R}=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}$$

Write a MATALB script to ask the user for the values R1, R2 and R3 and then write:

The total resistance is: xxx Ohms.

To the screen.

h) Write a program to load the file mars_data.dat, from example sheet 3. The program should print the size of the array into which it loads mars_data.dat. It should then ask the user to enter the region of array which it would like your program to plot. After the user has entered four values, x, xx, y and yy. A new array should be formed containing only the region the user is interested in. This should then be displayed using the **surf** command.