MATLAB

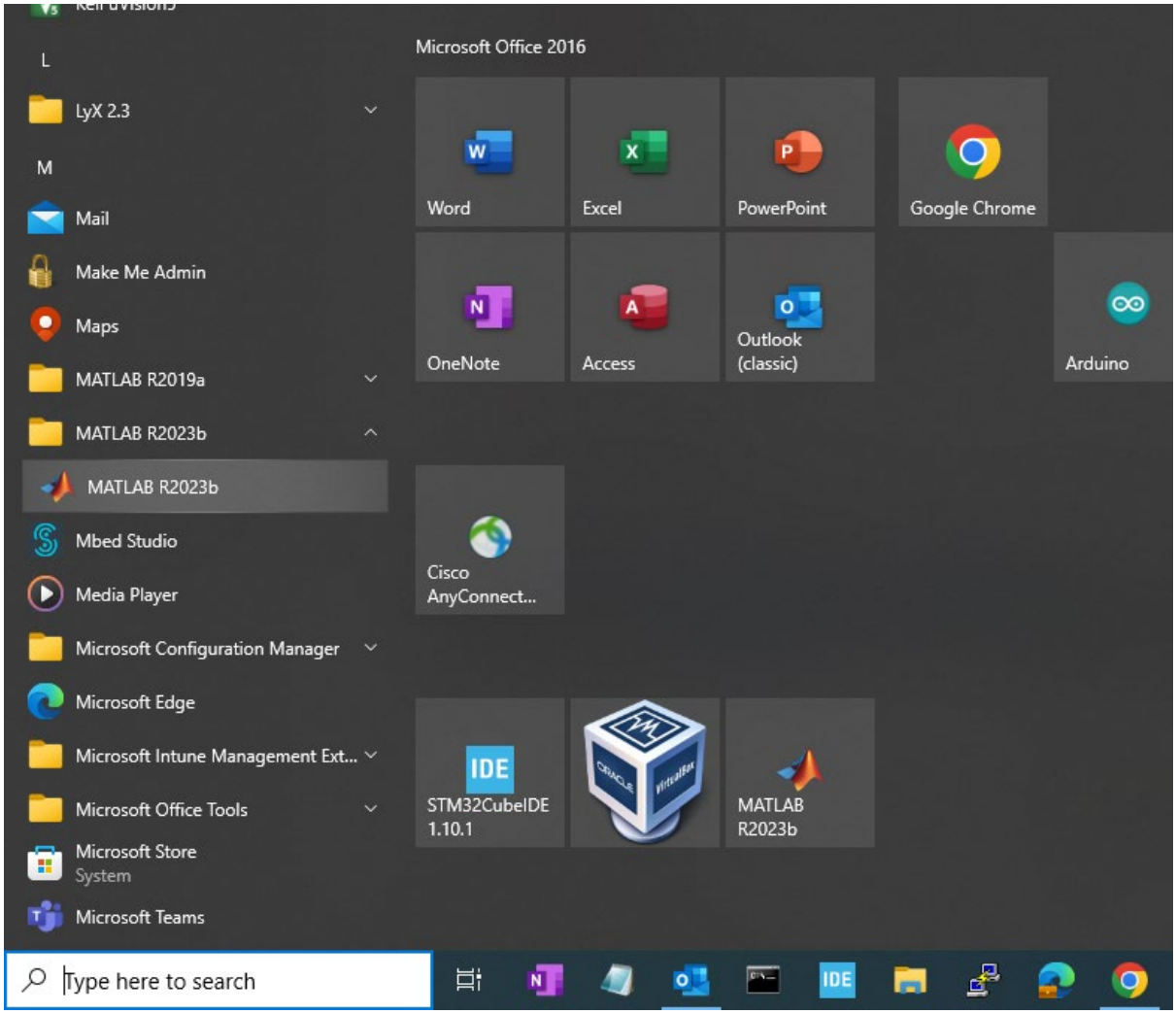
In labs 5 and 6 we will be using MATLAB to break weakened forms of popular encryption systems. We will only be using a small subset of the capabilities of MATLAB but for students who have not used it, this sheet is intended to introduce you to it. It contains all the MATLAB knowledge you need to complete the remaining labs.

You do not have to do so, but you can download MATLAB and use it on your own computer. You will need to install it using Swinburne's site license. You can also run a cloud based version. To do either of these you will need to set up a Mathworks account. Details are at https://au.mathworks.com/academia/tah-portal/swinburne-university-of-technology-30596663.html

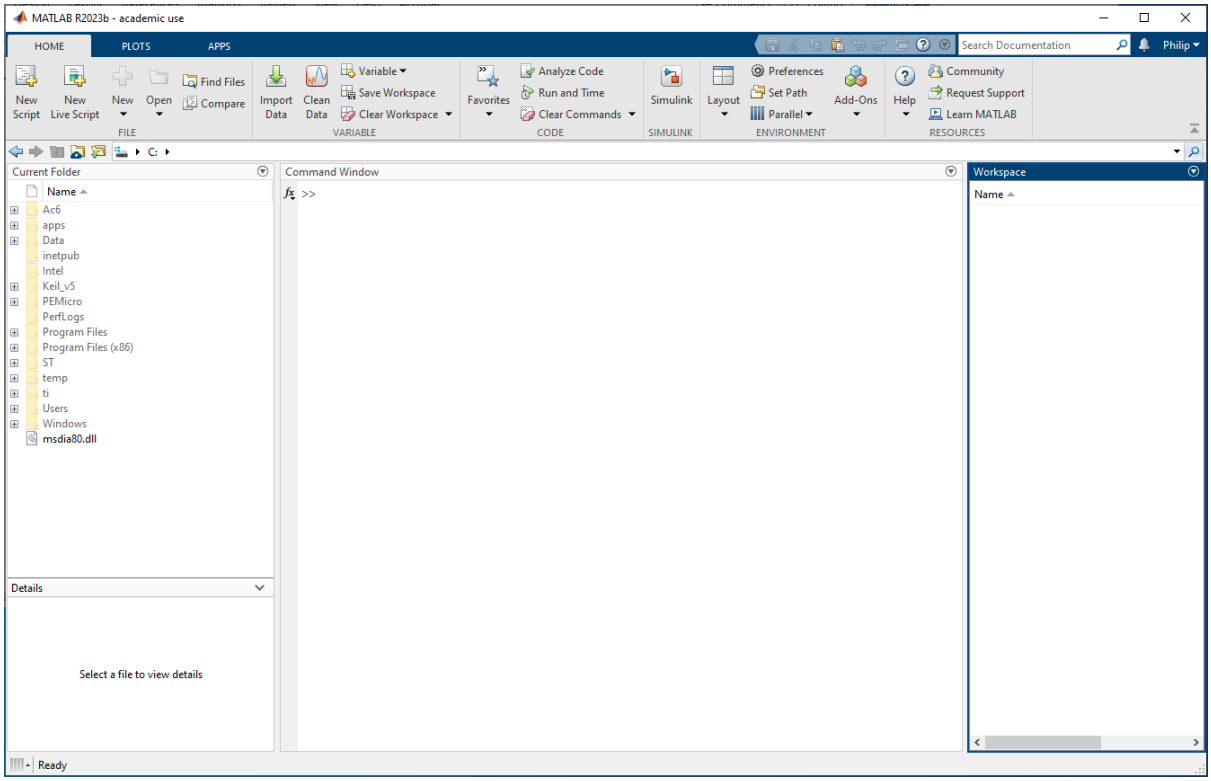
This sheet applies to the downloaded version of MATLAB but both the downloaded and cloud versions are very similar.

MATLAB overview

To start MATLAB find it in the start directory and start as usual.



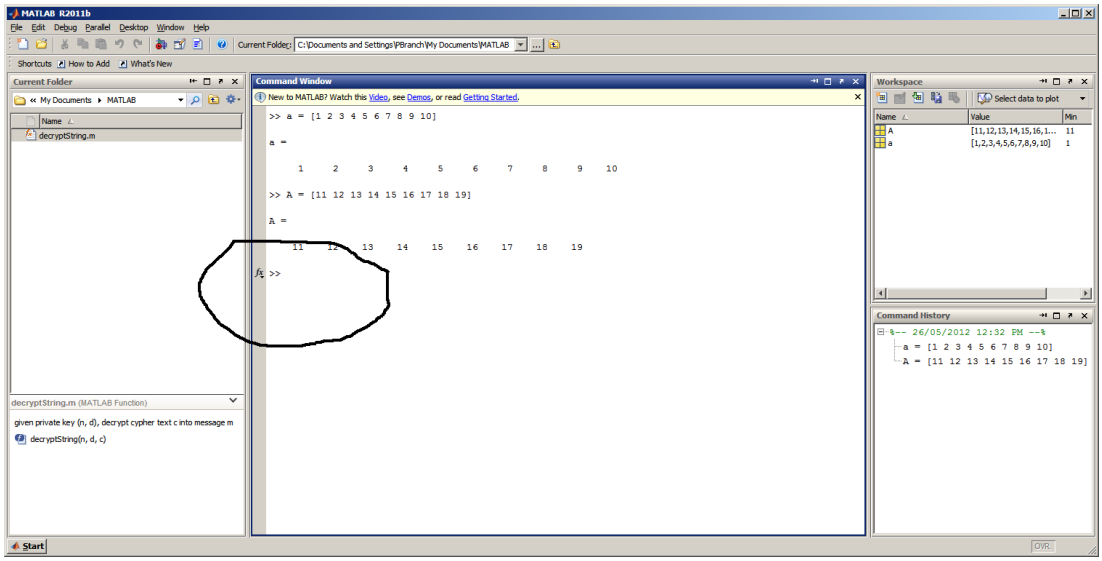
When started you will see the following:



The command window in the middle is used to run MATLAB instructions. The Workspace lists current variables and their values. Matlab instructions are typed or pasted into the command window. It is good practice to write your code into a text file and then paste it into the command window. The panel on the left lists directories that you can access files from. You can use the "Layout" tab to show the command history which I have done in the following screen shots. The Command History window keeps a record of all the instructions entered.

MATLAB is designed primarily to operate on matrices and vectors. We only need to deal with simple operations on vectors in this unit.

Commands in MATLAB are entered in the command window as below. I have defined two vectors `a` and `A`. Note that both are listed in the Workspace panel on the upper right hand side.



Vector definition

The simplest way to define a vector is to list its elements in order

Try `a = [ 1 2 3 4 5 6 7 8 9]`

You can suppress the listing of the vector by adding a semi-colon at the end. Also note that MATLAB is case sensitive.

Now try `A = [ 11 12 13 14 15 16 17 18 19];`

Accessing vector elements

Individual elements of a vector `A` are accessed by `A(i)`

Try `A(7)`

Putting a semicolon after a command suppresses output.

Try `a;`

Now try `a`

Displaying values

`disp(x)` displays the value of `x`

Try `disp(A)` and `disp(A(2))`

'for' loop

for loops in MATLAB can be implemented with

```
for count = start value : end value
   statement
end
```

Try

```
for i = 1:20
  x(i)= i;
  disp(x(i))
end;
```

`disp(x)` displays the value of `x`

Strings in MATLAB

Strings of characters can be defined in MATLAB with the ' delimiter.

Try `textstring = 'a string of text'`

Individual elements of the string can be accessed with the number of the element (starting from 1) in parentheses.

Try `textstring(5)`

Useful MATLAB commands

Along with constructing vectors, the following are the only commands you will need for the remaining labs:

`factor(n)` `returns the prime factors of n`

`for loop` `for i = 1:20 x(i)= i; end`

`if statement` `if (x==1) disp(x)`

`mod(x, y)` `returns x mod y`

`length(x)` `returns the length of a vector x`

`break` `ends execution of current for loop`

`disp(x)` `displays the value of x`