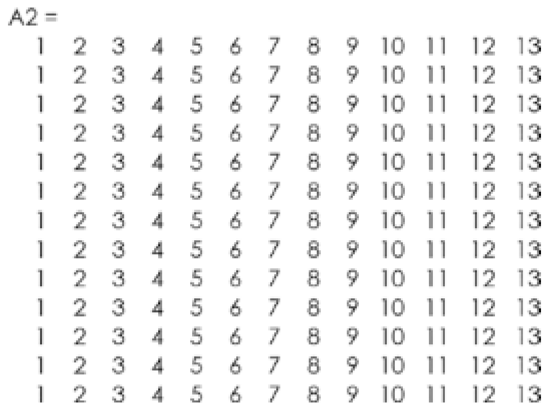
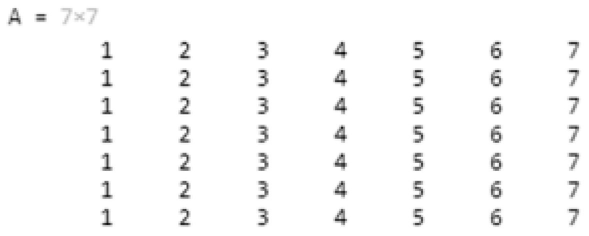
1. 1.Create the next array in the most efficient way (at least in two different ways). Note the display format of A2 elements



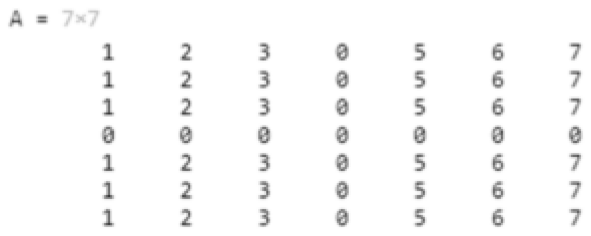
```
A2 =
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
  1  2  3  4  5  6  7  8  9  10  11  12  13
```

2. A5 = [1 2 3; 4 5 6; 7, 8, 9]
   a. Obtain B5 from A5 by two arithmetic operations: B5 = [16, 9 4; 1 0 1; 4 9 16].
   b. Obtain C5 from A5 and B5 by using relational logic (<, >) and arithmetic operations (+ 13): C5 = [13 13 13; 14 14 14; 14 13 13]
3. Create the matrix A in the most efficient way:



```
A = 7×7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
     1     2     3     4     5     6     7
```

   a. Obtain a new A matrix in a most efficient way:



```
A = 7×7
     1     2     3     0     5     6     7
     1     2     3     0     5     6     7
     1     2     3     0     5     6     7
     0     0     0     0     0     0     0
     1     2     3     0     5     6     7
     1     2     3     0     5     6     7
     1     2     3     0     5     6     7
```

4. Obtain the logical array C = [1 1 1 1 1 0 0 0 0 0 0] from the array A = [-5, -4, -3 … 3, 4, 5] whose elements are linear (equally spaced).
5. Obtain the numerical array D = [3 5 0; 0 0 4] from the array E = [3 5 7; 1 1 4].
6. x=linspace(-13, 0), F(x)=2*x^2+2*x-1;
   a. There are two potential errors. Find errors and fix them. What is the size of x and F now?
7. Create an * .m file called ARRAY_1.m and write the following:
   a. Write the command to clean up the Workspace and Command windows of MATLAB, and then display the current date and time in the Command window.
   b. Create array A. Write in it the commands to generate the following arrays: A1 (1-by-10) with the operator :, A2 (10-by-1) with linspace(), A3 (2-by-10) with eye().
   c. Create array B. Write in it the commands to create the following arrays: B1 (5-by-6) with randi() elements ranging between [-1….1], B2 (5-by-6) with rand(), and B3 (5-by-10) with randn().
   d. Create array B. Write in it the commands to create the following arrays: B1 (5-by-6) with randi() elements ranging between [-1….1], B2 (5-by-6) with rand(), and B3 (5-by-10) with randn().

e. Write in it the commands performing all possible (arithmetic array) operations (+, -, *, /, .*, ./, ^, .^) with A1, A2, and A3 (at least three operations) and call these new matrices: A1new1, A1new2, A1new3, A2new1, A2new2, A2new3, A3new1, A3new2, A3new3. Hints: use transpose() and rot90() while performing arithmetic array operations
f. Write in it the commands performing all possible (arithmetic array) operations (+, -, *, /, .*, ./, ^, .^, sum, mean) with B1, B2 and B3 (at least three operations) and call new matrices: B1new1, B1new2, B1new3, B2new1, B2new2, B2new3, B3new1, B3new2, B3new3. Hint: use fliplr() and transpose() while performing arithmetic array operations
g. Create AB1, AB2, and AB3 matrices from A1, A2, A3, and B1, B2, and B3. Also, use part of any A1, A2, A3 and B1, B2, B3 arrays. Note that every AB1, AB2, AB3, ABC4, ABC5 should contain some elements from arrays A and B. Hint: use flipud() and repmat() while creating the arrays AB1, AB2, and AB3
h. Create ABC1, ABC2, and ABC3 matrices by combining/concatenating the previously created arrays: A1, A2, A3 and B1, B2, B3 and C1, C2, C3. You should also use part of any A1, A2, A3 and B1, B2, B3 and C1, C2, C3 arrays. Note that every ABC1, ABC2, ABC3 should contain some elements from the A, B and C matrices from Parts 1, 2, and 3.

8. Write an M-file using the [while, end] loop control statement to compute the values of the cosine function g(θ) = cos (θ) for θ = 0...π with 2,000 incremental steps in this range and stop computation when the value of the function g(θ) ≈ 0.99999. Also, display how many steps it takes to get through the computation process and plot your simulation results. Also, display the end value of 0.99999 in the same plot.

9. Compute the area of a circle, square, and rectangle with regard to these user entries: W = input('Width of a rectangle: '); L = input('Length of a rectangle: '); R = input('Radius of a circle: '); S = input('Side length of a square: '); There are several scenarios to consider in your script: (1) if a user enters two dimensions for the width (W) and length (L) of a rectangle, your code has to compute the area of a rectangle and display it with comments; (2) if a user enters two dimensions, such as radius (R) and side (S) of a square, it has to compute the area of a circle and square, respectively, and display it with comments; (3) if a user enters all the required entries, it has to compute three areas and display with adequate comments; (4) if a user enters not all entries or misses any of the required entries (W, L, R, S), your code has to display "You need to ENTER all dimensions!". The input(), isempty(), exist(), length(), numel(), size(), and fprintf() commands can be employed.

10. Compute the series using the [while, end] and [for, end] loop control statements. Take n = 101 and plot error values over iterations.

$$\frac{\pi^2}{8} = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \ldots = \sum_{n=0}^{\infty} \frac{1}{(2n+1)^2}$$

11. Compute the sum of these series: 1 ∗ 2 + 2 ∗ 3 + 3 ∗ 4 + 4 ∗ 5 + ... , Plot the sum as a function of k.This can be rewritten as a sum

$$\sum_{k=1}^{N} k(k+1).$$

12. Compute these series (discovered by Euler in 1735). Take n = 1...99 and plot error values over iterations. Run the exhaustive computation and halt the computation process when the error is

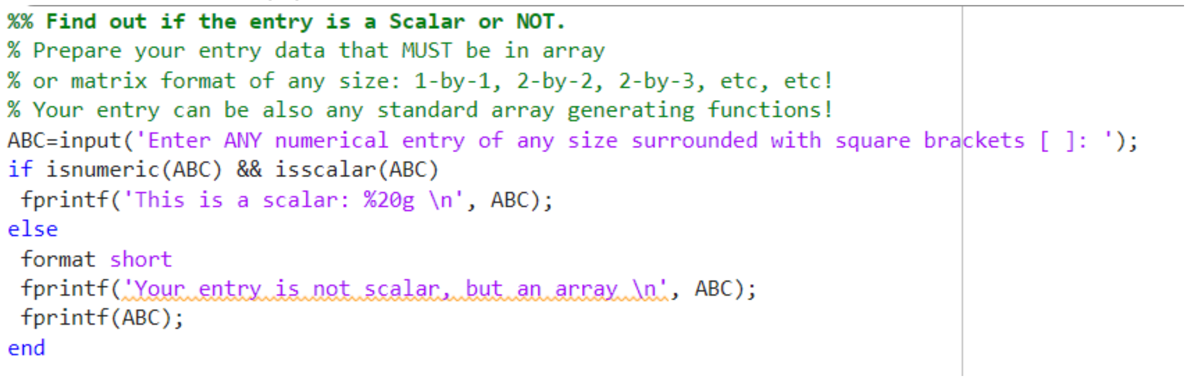
smaller than 10e−13.

$$\frac{\pi^2}{6}=1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\ldots=\sum_{n=1}^{\infty}\frac{1}{n^2}$$

13. Compute this series, Take n = 11, bn = [−5...5], x = [0...L], and L = 10. For x take a step size of Plot all computation results. $\Delta x=\frac{L}{100}$.

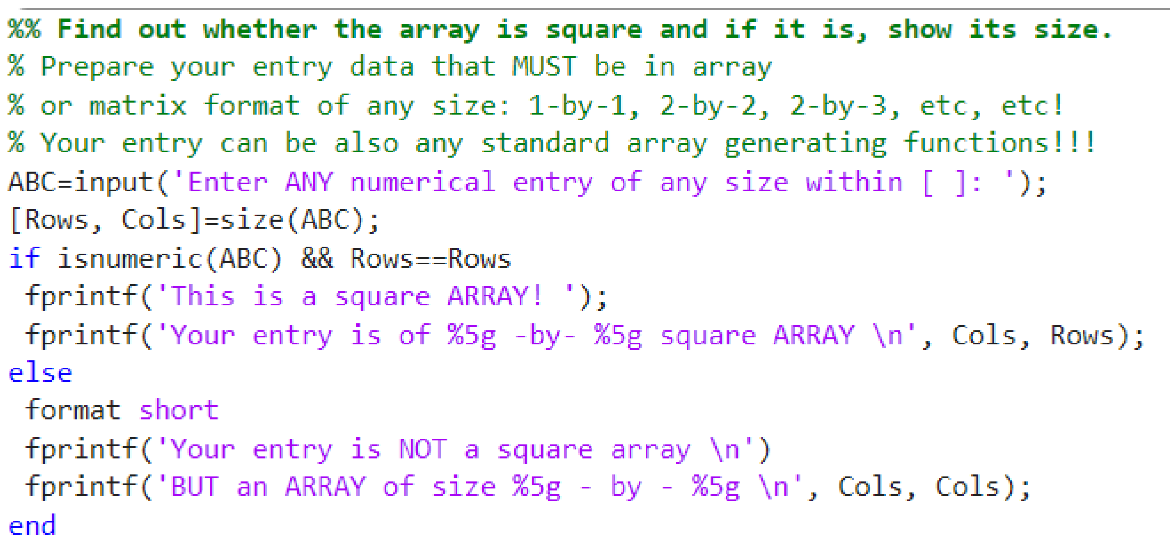
$$f(x)=\sum_{n=1}^{\infty}\frac{b_n sin\left((2n-1)\pi x\right)}{2L}$$

14. Fix two errors in the following given script



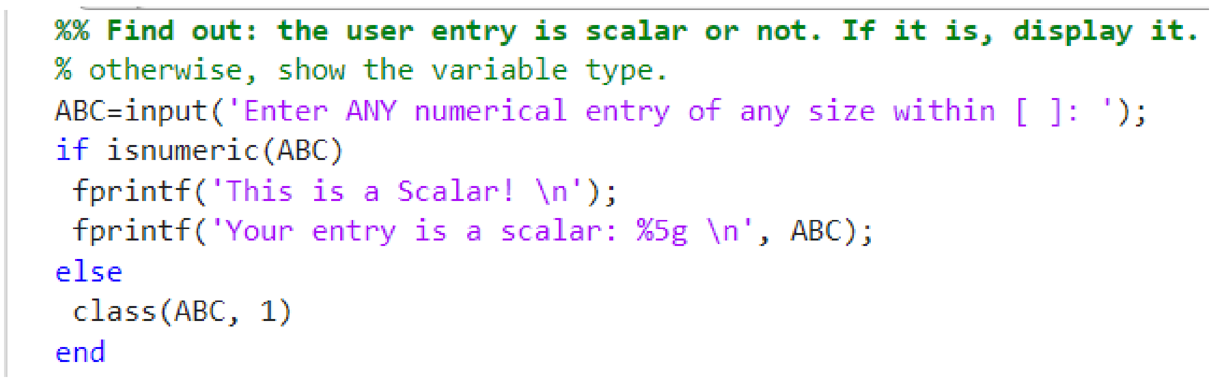
```
%% Find out if the entry is a Scalar or NOT.
% Prepare your entry data that MUST be in array
% or matrix format of any size: 1-by-1, 2-by-2, 2-by-3, etc, etc!
% Your entry can be also any standard array generating functions!
ABC=input('Enter ANY numerical entry of any size surrounded with square brackets [ ]: ');
if isnumeric(ABC) && isscalar(ABC)
 fprintf('This is a scalar: %20g \n', ABC);
else
 format short
 fprintf('Your entry is not scalar, but an array \n', ABC);
 fprintf(ABC);
end
```

15. Fix the three errors in the following script



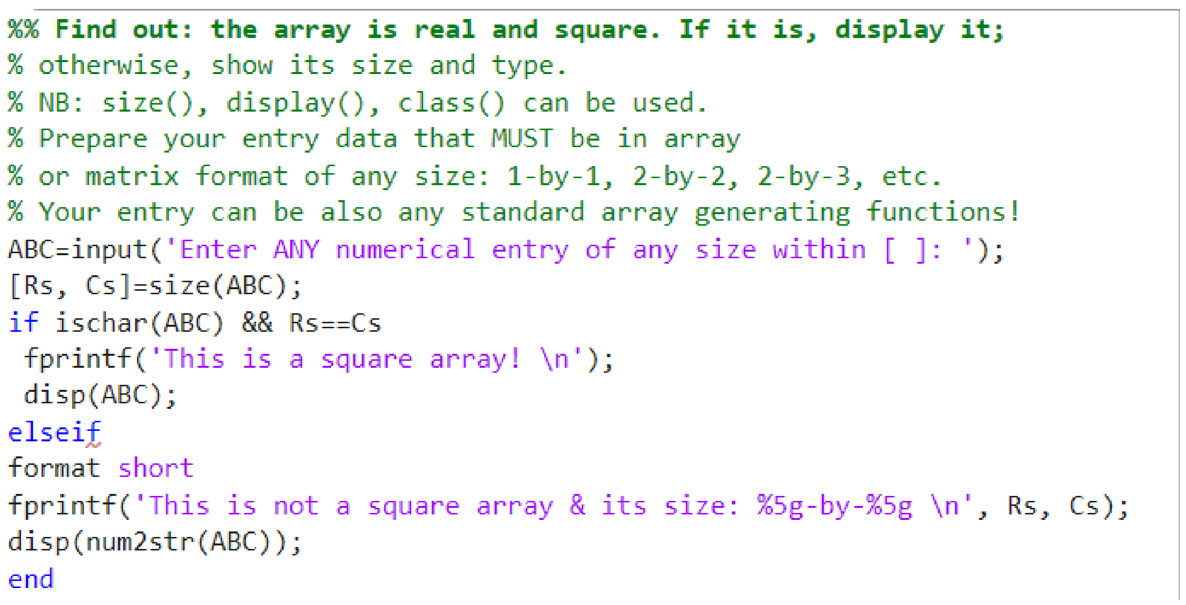
```
%% Find out whether the array is square and if it is, show its size.
% Prepare your entry data that MUST be in array
% or matrix format of any size: 1-by-1, 2-by-2, 2-by-3, etc, etc!
% Your entry can be also any standard array generating functions!!!
ABC=input('Enter ANY numerical entry of any size within [ ]: ');
[Rows, Cols]=size(ABC);
if isnumeric(ABC) && Rows==Rows
 fprintf('This is a square ARRAY! ');
 fprintf('Your entry is of %5g -by- %5g square ARRAY \n', Cols, Rows);
else
 format short
 fprintf('Your entry is NOT a square array \n')
 fprintf('BUT an ARRAY of size %5g - by - %5g \n', Cols, Cols);
end
```

16. Fix the two errors in the following script



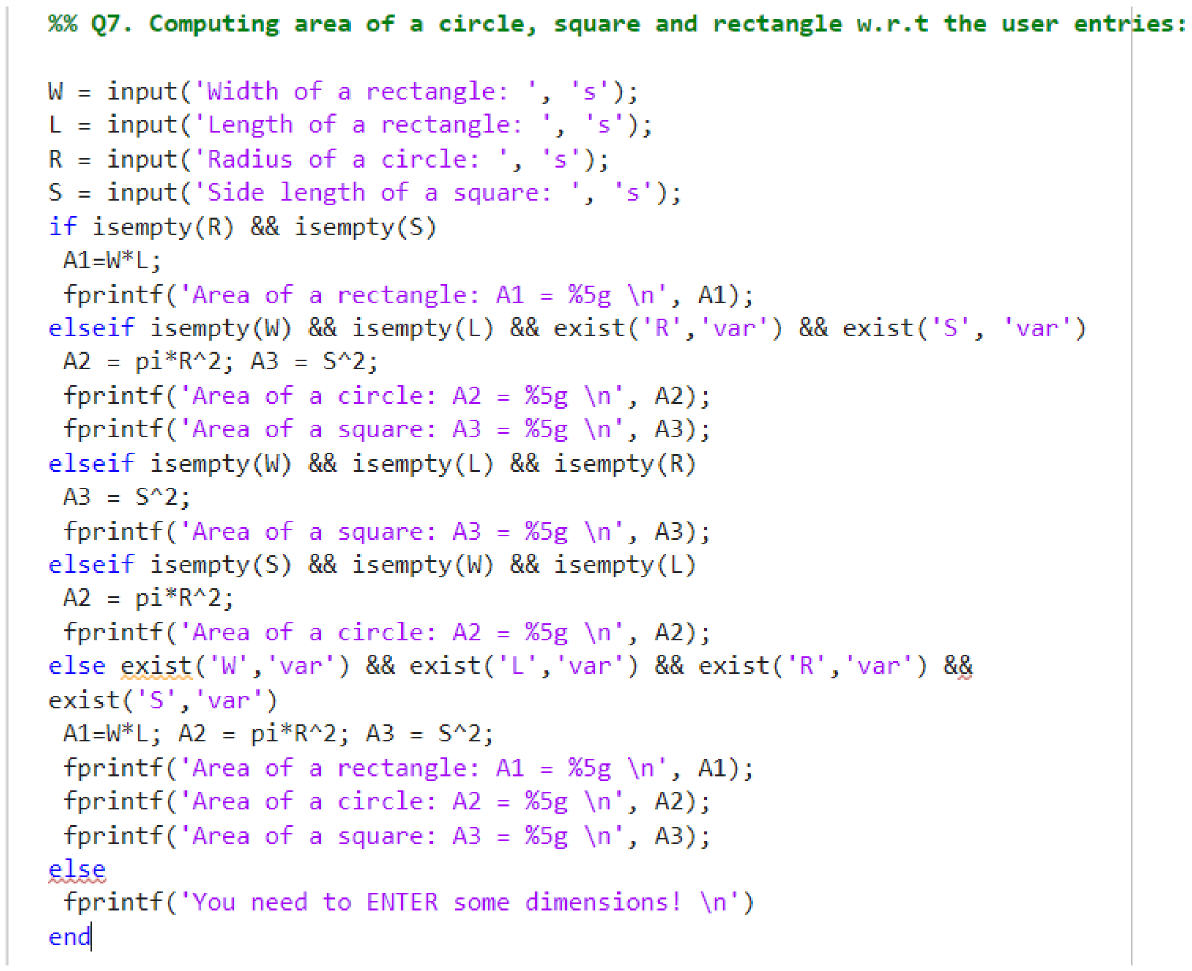
```
%% Find out: the user entry is scalar or not. If it is, display it.
% otherwise, show the variable type.
ABC=input('Enter ANY numerical entry of any size within [ ]: ');
if isnumeric(ABC)
 fprintf('This is a Scalar! \n');
 fprintf('Your entry is a scalar: %5g \n', ABC);
else
 class(ABC, 1)
end
```

17. Fix two errors in the following script



```
%% Find out: the array is real and square. If it is, display it;
% otherwise, show its size and type.
% NB: size(), display(), class() can be used.
% Prepare your entry data that MUST be in array
% or matrix format of any size: 1-by-1, 2-by-2, 2-by-3, etc.
% Your entry can be also any standard array generating functions!
ABC=input('Enter ANY numerical entry of any size within [ ]: ');
[Rs, Cs]=size(ABC);
if ischar(ABC) && Rs==Cs
 fprintf('This is a square array! \n');
 disp(ABC);
elseif
format short
fprintf('This is not a square array & its size: %5g-by-%5g \n', Rs, Cs);
disp(num2str(ABC));
end
```

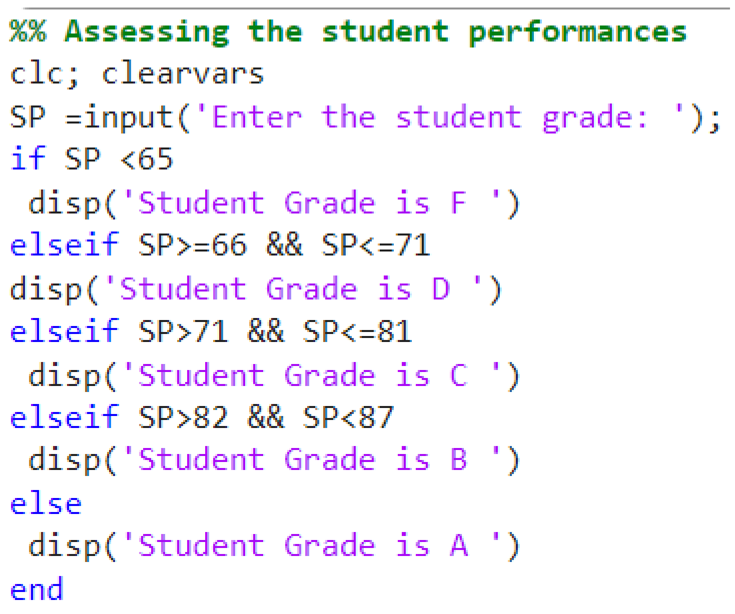
18. Fix the five errors in the following script:



```
%% Q7. Computing area of a circle, square and rectangle w.r.t the user entries:

W = input('Width of a rectangle: ', 's');
L = input('Length of a rectangle: ', 's');
R = input('Radius of a circle: ', 's');
S = input('Side length of a square: ', 's');
if isempty(R) && isempty(S)
 A1=W*L;
 fprintf('Area of a rectangle: A1 = %5g \n', A1);
elseif isempty(W) && isempty(L) && exist('R','var') && exist('S', 'var')
 A2 = pi*R^2; A3 = S^2;
 fprintf('Area of a circle: A2 = %5g \n', A2);
 fprintf('Area of a square: A3 = %5g \n', A3);
elseif isempty(W) && isempty(L) && isempty(R)
 A3 = S^2;
 fprintf('Area of a square: A3 = %5g \n', A3);
elseif isempty(S) && isempty(W) && isempty(L)
 A2 = pi*R^2;
 fprintf('Area of a circle: A2 = %5g \n', A2);
else exist('W','var') && exist('L','var') && exist('R','var') &&
exist('S','var')
 A1=W*L; A2 = pi*R^2; A3 = S^2;
 fprintf('Area of a rectangle: A1 = %5g \n', A1);
 fprintf('Area of a circle: A2 = %5g \n', A2);
 fprintf('Area of a square: A3 = %5g \n', A3);
else
 fprintf('You need to ENTER some dimensions! \n')
end
```

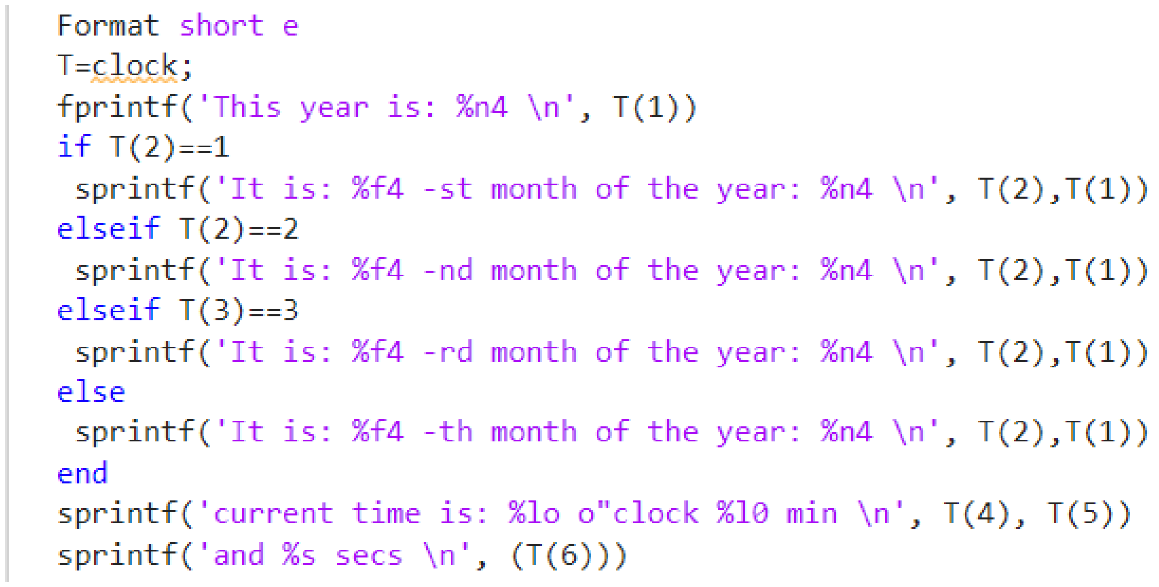
19. Fix the five errors in the following script



```
%% Assessing the student performances
clc; clearvars
SP =input('Enter the student grade: ');
if SP <65
 disp('Student Grade is F ')
elseif SP>=66 && SP<=71
disp('Student Grade is D ')
elseif SP>71 && SP<=81
 disp('Student Grade is C ')
elseif SP>82 && SP<87
 disp('Student Grade is B ')
else
 disp('Student Grade is A ')
end
```

20. Write a script (program) that computes the volume and weight of the model that may have a form of cube, cylinder, and rectangular prism. Users need to enter (via input prompt) the necessary geometric dimensions of the model and enter or select material properties (density) from the given data (aluminum, copper, and steel) in your script. Your script has to write all computed and user input data sets into an external file called RESULTS.txt with explanatory comments in it along with numerical data.

21. Edit and correct the following given script to display the current date and time correctly in the Command window:



```
Format short e
T=clock;
fprintf('This year is: %n4 \n', T(1))
if T(2)==1
 sprintf('It is: %f4 -st month of the year: %n4 \n', T(2),T(1))
elseif T(2)==2
 sprintf('It is: %f4 -nd month of the year: %n4 \n', T(2),T(1))
elseif T(3)==3
 sprintf('It is: %f4 -rd month of the year: %n4 \n', T(2),T(1))
else
 sprintf('It is: %f4 -th month of the year: %n4 \n', T(2),T(1))
end
sprintf('current time is: %lo o"clock %l0 min \n', T(4), T(5))
sprintf('and %s secs \n', (T(6)))
```

22. Your corrected script should display the current date and time in the following format:
    It is: 11 - day of the 6-th month of the year: 2014
    current time is: 15 o"clock 3 min and 17.136 secs

23. Create a function file taking one input variable (k) and one output variable (S). Compute a total sum of these series:

$$10+35+65+110+\ldots=\sum_{m=0}^{55}\left(10*2.5^{k}\right).$$

a. Halt the computation process when the sum is larger than 2.5 ∗ 1018 and display in the Command window at what iteration step the computation is halted and the computed sum of the series

b. Display the difference between the computed sum of series and 2.5 ∗ 1018.

c. Display the final sum and the iteration number as integers, but collect all of the sums (S) from every step.