

1 Problem 1

```
clear;clc;
% Main Program %
% numerically integrate  $y(x) = ax^2 + bx + c$  from lowerlimit to upperlimit
a = 2;
b = 2;
c = 1;
% from 0 to 1
lowerlimit = 0;
upperlimit = 1;
% number of discretizations
N = 1000;
% choice: 1 = Riemann, 2 = Trapezoidal
choice = 2;fprintf('Using method %i, the area is: ',choice)
disp( intpoly(a,b,c,lowerlimit,upperlimit,N,choice) )

% Functions %

function [Area] = intpoly(a,b,c,lowerlimit,upperlimit,N,choice)
% This function integrates a specific polynomial (y(x)) using the
% Trapezoidal Rule or Left Riemann Sum
x = linspace(lowerlimit,upperlimit,N);
y = a.*x.^2 + b.*x + c;
if choice == 2
    A = (upperlimit - lowerlimit)/(2*N); %Trap
    P = y(1,2:end-1);
    Area1 = A*(y(1)+y(end));
    Area2 = sum(P)*A*2;
    Area = Area1+Area2;
elseif choice == 1
    A = (upperlimit - lowerlimit)/N;
    P = y(1,2:end-1);
    Area1 = (y(1)+y(end-1))*A;
    Area2 = (sum(P)*A);
    Area = Area1 + Area2;
else
    msg = 'Choice needs to be 1 or 2!';
    error(msg)
end
end
% If I did not want to use the sum command, then I would have
% to use a loop to sum all the elements of my array
```

2 Problem 2

```
clear;clc;

x0(1) = 0;
y0(1) = 0;
v0 = 20; % initial speed in m/s
angle = 45; % elevation angle in degrees
[maxheight,time,speed] = cannon(x0,y0,v0,angle);
fprintf('The max height is %7.2f meters\n' , maxheight)
fprintf('The time of the max height is %7.2f seconds \n' , time)
fprintf('The speed at the max height is %7.2f m/s\n' , speed)

%Function
function [maxheight,time,speed] = cannon(x0,y0,v0,angle)
%This function calculates the max height, time at which it occurs, and the
%speed at that time
Count = 1;
T(1) = 0;
Theta = (angle*pi)/180;
g = -9.8;
V0x = v0*cos(Theta);
V0y = v0*sin(Theta);
while y0>=0
    Count = Count + 1;
    T(Count) = T(Count - 1) + .05;
    y0(Count) = V0y.*T(Count) + (.5*g).*T(Count).^2;
    x0(Count) = V0x.*T(Count);
end
Maxy0 = max(y0);
[row,col] = find(y0 == Maxy0);
maxheight = y0([row],[col]);
time = T([row],[col]);
Max_Vx2 = V0x^2;
Vy2 = (V0y^2) + (2*g).*(y0);
Max_Vy2 = Vy2([row],[col]);
speed = sqrt(Max_Vx2 + Max_Vy2);
end
```

3 Problem 3

```
clear;clc;

n=5;
magicarray = magicsq(n)

% Function

function [Result] = magicsq(n)
Matrix = zeros(n);
row = 1;
col = (n+1)/2;
for k = 1:(n^2)
    Matrix(row,col) = k;
    row = row - 1;
    col = col+ 1;
    if row == 0 && col <= n
        row = n;
    elseif row > 0 && col > n
        col = col - n;
    elseif (row == 0 && col > n) || Matrix(row,col) > 0
        row = row + 2;
        col = col - 1;
    end
end
Result = Matrix;
end
```