ENG10004

Digital and Data Systems

Portfolio

Student Name : Joshua Lillington-Moore

Student ID : 103666887

Date :

Self Assessment Details :

	Fail	Pass	Credit	Distinction	High Distinction
Self Assessment (pls tick)		🟩	🟩		

Pass Borderline Tasks Checklist :

	PB1	PB2	PB3	PB4	PB5	PB6
	🟪	🟪	🟪	🟨	🟪	🟪
	🟩	🟩	🟩	🟩	🟩	🟩

Pass Plus Tasks Checklist :

	PP1	PP2
	🟪	🟪
	🟩	🟩

Credit Tasks Checklist :

	C1	C2
	🟪	🟨
	🟩	

Distinction Tasks Checklist :

	D1	D2

High Distinction Tasks Checklist :

	HD1	HD2

Declaration :

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature :

Pass Borderline :

Task 1 :

Problem 1 :

1)

```
>> a=16

a =

    16
```

2)

```
>> b=2.4*10^25

b =

   2.4000e+25
```

3)

```
>> c=3+6i

c =

   3.0000 + 6.0000i
```

Problem 2 :

1)

```
>> x=2/(1+exp(-(a-10)/5))

x =
    1.5370
```

2)

```
>> y=(sqrt(a)+b^(1/3))^pi

y =

   3.7853e+26
```

3)

```
>> z=tan(real(c))+sin(pi/5)^2

z =

    0.2029
```

4)

```
>> w=log(abs(a^2.5-1202))

w =

    5.1818
```

Problem 3 :

1)

```
>> aVec= [17 ; 3.14 ; 7 ; 23]

aVec =

   17.0000
    3.1400
    7.0000
   23.0000
```

2)

```
>> bVec=[2.51,5,28,170]

bVec =
    2.5100    5.0000   28.0000  170.0000
```

3)

```
>> cVec = 2:-0.5:-0.5

cVec =

    2.0000    1.5000    1.0000    0.5000         0   -0.5000
```

4)

```
>> dVec = [10^0.01 log(abs(49)) 975 sqrt(a)]'

dVec =

    1.0233
    3.8918
  975.0000
    4.0000
```

5)

```
>> eVec = 'Hello Swinburne'

eVec =

    'Hello Swinburne'
```

Problem 4 :

1)

```
>> xVec = (1/sqrt(2*pi))*exp(-cVec.^2/2.5)

xVec =

    0.0805    0.1622    0.2674    0.3610    0.3989    0.3610
```

2)

```
>> yVec = ((aVec).^2+bVec.^2).^1/2

yVec =

   1.0e+04 *
```

```
    0.0148    0.0157    0.0537    1.4594
    0.0008    0.0017    0.0397    1.4455
    0.0028    0.0037    0.0416    1.4474
    0.0268    0.0277    0.0657    1.4714
```

3)

```
>> zVec = log10(1./dVec)

zVec =

   -0.0100
   -0.5902
   -2.9890
   -0.6021
```

Problem 5 :

```
>> Value = 123.4567e2;
>> fprintf('Value = %e\n', Value);
Value = 1.234567e+04
>> fprintf('Value = %f\n', Value);
Value = 12345.670000
>> fprintf('Value = %g\n', Value);
Value = 12345.7
>> fprintf('Value = %12.4f\n', Value);
Value =   12345.6700
```

2) Compare Outputs :

Task 2 :

Problem 1 :

1)

```
>> aMat = [1 -3 ; 4 6]

aMat =

     1    -3
     4     6
```

2)

```
>> bMat = ones(3,4)*5
```

```
bMat =

     5     5     5     5
     5     5     5     5
     5     5     5     5
```

3)

```
>> cMat = [aMat aMat ; aMat aMat]

cMat =

     1    -3     1    -3
     4     6     4     6
     1    -3     1    -3
     4     6     4     6
```

4)

```
>> dMat = [10:2:16; [4 3 6 8]; zeros(1,2) 4 9; -1:-3:-10]

dMat =

    10    12    14    16
     4     3     6     8
     0     0     4     9
    -1    -4    -7   -10
```

Problem 2 :

1)

```
>> xMat=aMat^2

xMat =

   -11   -21
    28    24
```

2)

```
>> yMat = aMat.^2

yMat =

     1     9
    16    36
```

‘^2’ is a matrix operation while ‘.^2’ multiplies elements.

3)

```
>> zMat = cMat'+inv(dMat)

zMat =

    1.0848    4.1250    1.1250    4.3482
   -2.6875    5.2500   -2.7500    6.1250
    0.4777    5.1250    0.1250    3.2768
   -2.7679    5.5000   -2.5000    6.3214
```

Problem 3 :

1)

```
>> size(aMat)

ans =

     2     2
```

2)

```
>> bMat(1,:)=[1 1 1 1]

bMat =

     1     1     1     1
     5     5     5     5
     5     5     5     5
```

3)

```
>> cMat(1:end,[2])

ans =

    -3
     6
    -3
     6
```

4)

```
>> dMat (1:3,2:4)

ans =
  12    14    16
   3     6     8
   0     4     9
```

Problem 4 :

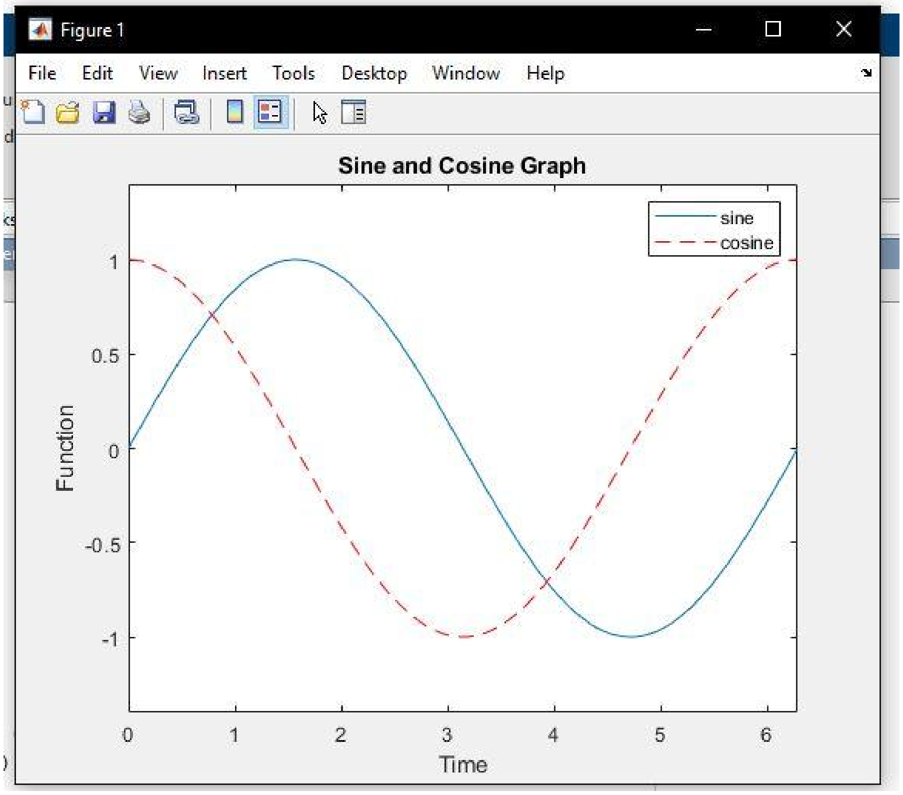
```
radius = input('Enter the circle radius:\n');
Area = pi*radius^2;
Perimeter = 2*pi*radius;
fprintf('The area is %f \n', Area);
disp(['The perimeter is ' num2str(Perimeter)]);
```

```
>> Task2Part4
Enter the circle radius:
30
The area is 2827.433388
The perimeter is 188.4956
```

The first statement asks the user to input the radius of the circle, this creates a new variable. The radius is then used to calculate the area and perimeter which is then printed and displayed for the user to see.

Problem 5 :

```
clear all;
clc;
close all;
figure(1)
t = 0:pi/100:2*pi;
x1=sin(t);
x2=cos(t);
plot(t,x1);
hold on;
plot(t,x2,'r--');
hold off;
xlim([0 2*pi])
ylim([-1.4 1.4])
xlabel('Time');
ylabel('Function');
title('Sine and Cosine Graph');
legend('sine','cosine')
```



Task 3 :

Problem 1 :

1)

After Debugging the program using a breakpoint. It can be seen that the errors are that there is a missing multiplication sign between v0 and t. There is a missing f at the end of ‘fprint’. Also a missing ‘ after the ‘/n’ and a missing ; after the bracket in the last line. The ‘⅓’ in line 8 should also be ‘½’. Instead of ‘x=x1-x2’ it should be ‘x=x1+x2’ and ‘a’ should equal ‘-9.81’ not ‘9.81’. After changing all of these it will give the correct value.

2)

```
t=5;
x0=95;
v0=17;
a=-9.81;

x1=x0+v0*t;
x2=1/2*a*t^2;

x=x1+x2;
fprintf('The position of the ball is: %f m.\n',x);
```

```
>> Task3Part1
The position of the ball is: 57.375000 m.
```

Problem 2 :

```
hours = input('Please enter the number of hours worked:\n');
pay = input('Please enter the hourly wage in $:\n');
if hours <= 38
   weeklypay = hours*pay
else
   weeklypay = hours*pay+(hours-38)*0.5*pay
end
disp(['The worker''s pay is ' num2str(weeklypay)]);
```

```
>> Task3Part2
Please enter the number of hours worked:
30
Please enter the hourly wage in $:
20

weeklypay =
   600

The worker's pay is 600

>> Task3Part2
Please enter the number of hours worked:
42
Please enter the hourly wage in $:
20

weeklypay =

   880

The worker's pay is 880
```

Problem 3 :

```
m = input('Please enter your mark (0 - 100):\n');
if m>100
   disp('Your grade is Invalid')
elseif m<0
   disp('Your grade is Invalid')
elseif m>=80
   disp('Your grade is High Distinction')
elseif m>=70
   disp('Your grade is Distinction')
elseif m>=60
   disp('Your grade is Credit')
elseif m>=50
   disp('Your grade is Pass')
elseif m<50
   disp('Your grade is Fail')
end
```

```
>> Task3Part3
Please enter your mark (0 - 100):
94
Your grade is High Distinction

>> Task3Part3
Please enter your mark (0 - 100):
60
Your grade is Credit
```

```
Please enter your mark (0 - 100):
110
Your grade is Invalid
```

Problem 4 :

```
n = input('Please enter a value for n : ');
sum=0;
for k=1:1:n
   x=(((-1)^k)*k^2)/1.3^(2*k+1);
   sum=sum + x;
end
disp(['The result for n is ' num2str(sum)]);
```

```
>> Task3Part4
Please enter a value for n : 4
The result for n is 0.69665

>> Task3Part4
Please enter a value for n : 20
The result for n is -0.042715
```

Problem 5 :

```
sum=0

while sum<=9
   x=randn();
   if x>=3
      disp('The number is greater than 3')
      break
   end
   sum = sum+x

end
fprintf('The total sum of the random numbers is: %f\n',sum)
fprintf('The last random number generated is: %f\n',x)
```

```
sum =

   8.5245
```

```
sum =

   11.0046

The total sum of the random numbers is: 11.004643
The last random number generated is: 2.480164

sum =

   -3.8135


sum =

   -3.6098

The number is greater than 3
The total sum of the random numbers is: -3.609773
The last random number generated is: 3.408806
```

Task 4 :

Problem 1 :

```
function y = PBTask4p1(x)
y = 3.2*x.^4-5*x.^2-7*x;
hold off ;

x1=[-5:0.1:5];
y1 = PBTask4p1(x1);
xlim([-5 5])
ylim([0 2000])
xlabel('Time');
ylabel('Function');
plot(x1,y1)
end
```

(doesn’t work)

Problem 2 :

```
function a = PBTask4p2a_f(x)
a=x;
global COUNTER %Declare COUNTER to be global.
```

```
COUNTER = COUNTER + x;
if (COUNTER > 1000);
    COUNTER = 0; %Specify initial value.
end
fprintf('COUNTER = %d', COUNTER)
end

function b = PBTask4p2b_f(x)
b=x;
global COUNTER %Declare COUNTER to be global.
COUNTER = COUNTER + x;
if (COUNTER > 1000);
    COUNTER = 0; %Specify initial value.
end
fprintf('COUNTER = %d', COUNTER)
end
```

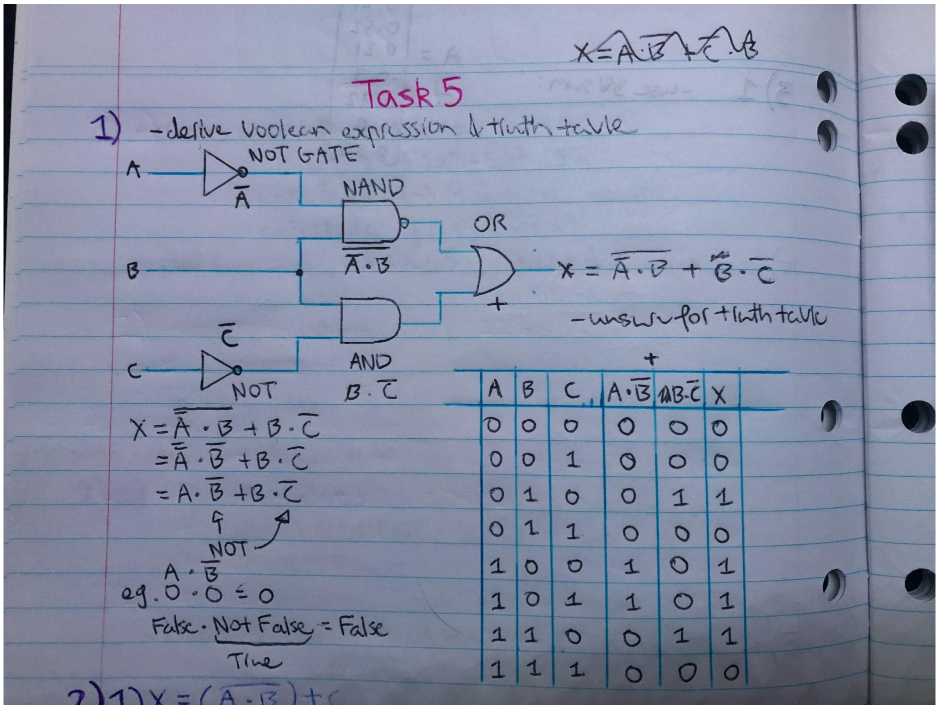
```
>> PBTask4p2a_f(200)
COUNTER = 200
ans =

   200

>> PBTask4p2b_f(600)
COUNTER = 800
ans =

   600

>> PBTask4p2a_f(300)
COUNTER = 0
ans =

   300
```

Problem 3 :

I don't know how to do this at all in a script.

Task 5 :

Problem 1 :



$X=A\cdot\overline{B}+\overline{C}\cdot B$

Task 5

1) -derive voolean expression & truth tavle

A
NOT GATE
$\overline{A}$
NAND
$\overline{\overline{A}\cdot B}$
B
OR
+
$\overline{C}$
C
NOT
AND
$B\cdot\overline{C}$

$X=\overline{\overline{A}\cdot B}+\overline{B}\cdot\overline{C}$

-answer for truth tavle

$X=\overline{\overline{A}\cdot B}+B\cdot\overline{C}$

$=\overline{\overline{A}}\cdot\overline{B}+B\cdot\overline{C}$

$=A\cdot\overline{B}+B\cdot\overline{C}$

&

NOT

$A\cdot\overline{B}$

eg. $0\cdot0=0$

$\text{False}\cdot\underbrace{\text{Not False}}_{\text{True}}=\text{False}$

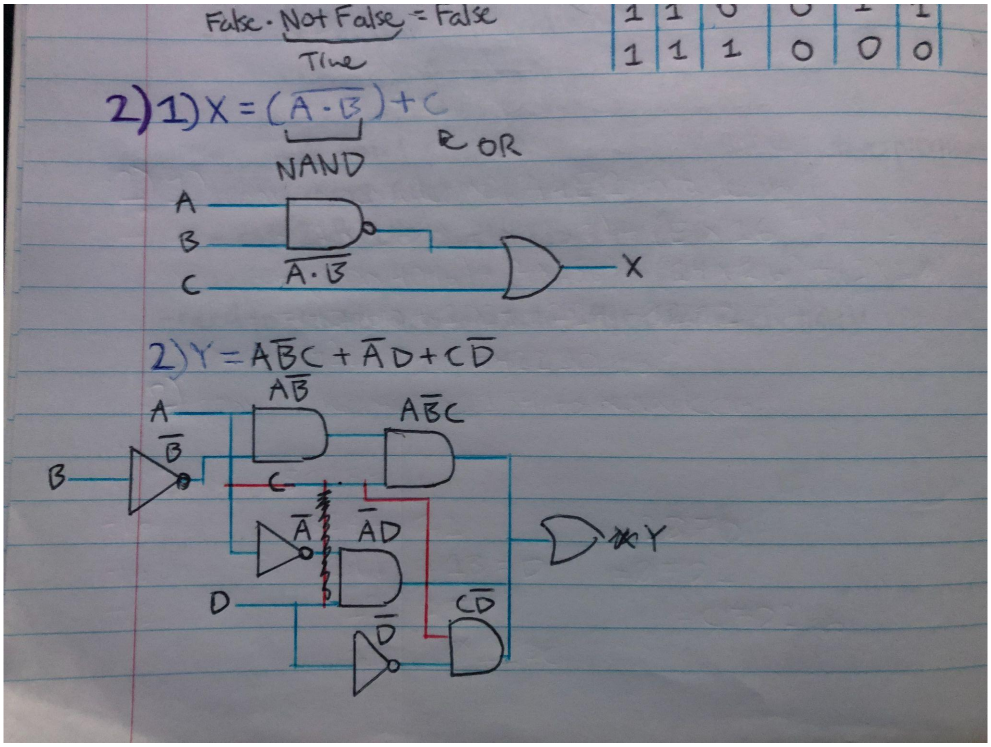
+

A	B	C	$A\cdot\overline{B}$	~~A~~$B\cdot\overline{C}$	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	0

2) 1) $X=(\overline{A\cdot B})+C$

Incorrect, needed to use De Morgan's Theorem and fix the truth table, ran out of time.

Problem 2 :

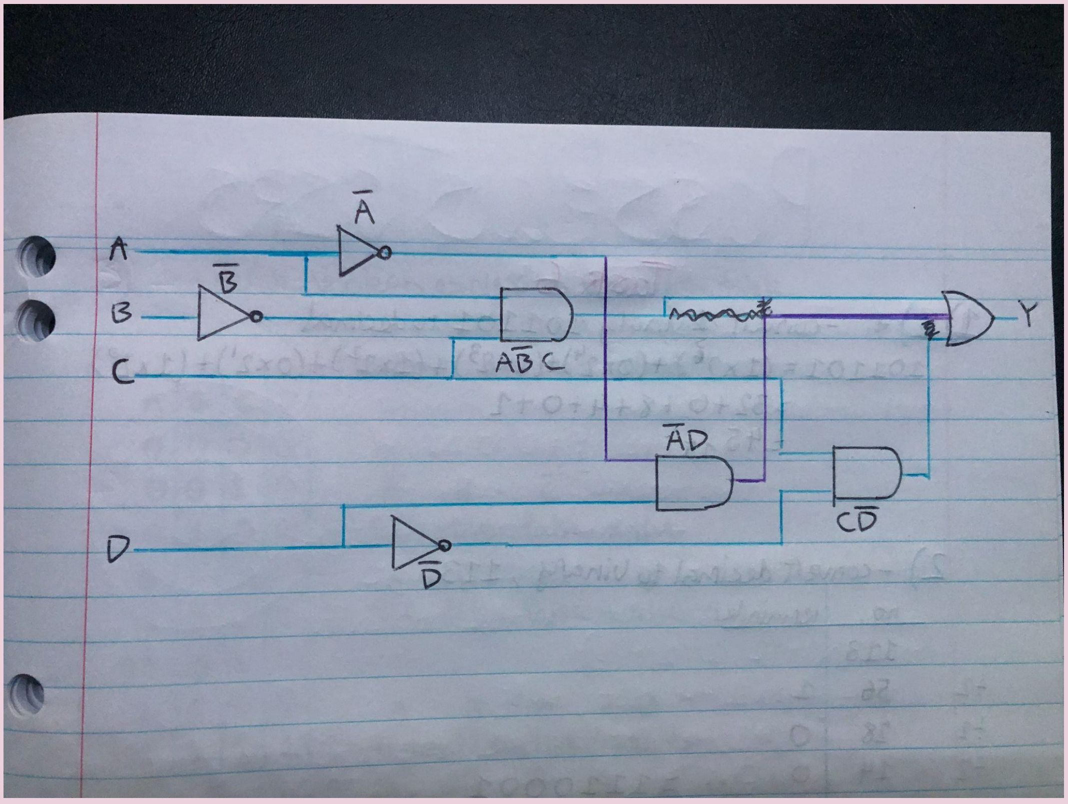


$\text{False} \cdot \underbrace{\text{Not False}}_{\text{True}} = \text{False}$

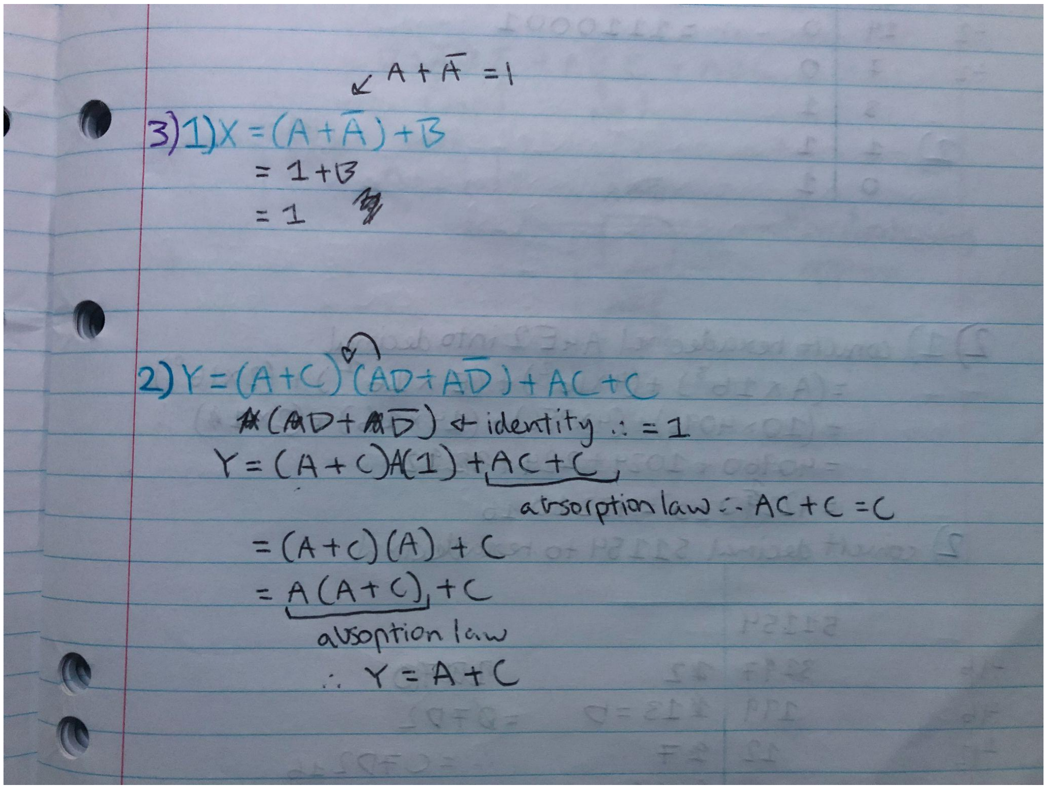
1	1	0	0	1	1
1	1	1	0	0	0

2) 1) $X = (\underbrace{\overline{A \cdot B}}_{\text{NAND}}) + C$ & OR

2) $Y = A\overline{B}C + \overline{A}D + C\overline{D}$



Problem 3 :



$\swarrow A + \bar{A} = 1$

3) 1) $X = (A + \bar{A}) + B$

$= 1 + B$

$= 1$

2) $Y = (A+C)(AD + A\bar{D}) + AC + C$

$(\cancel{A}D + \cancel{A}\bar{D})$ ← identity ∴ $= 1$

$Y = (A+C)A(1) + \underbrace{AC + C}_{\text{absorption law ∴ } AC + C = C}$

$= (A+C)(A) + C$

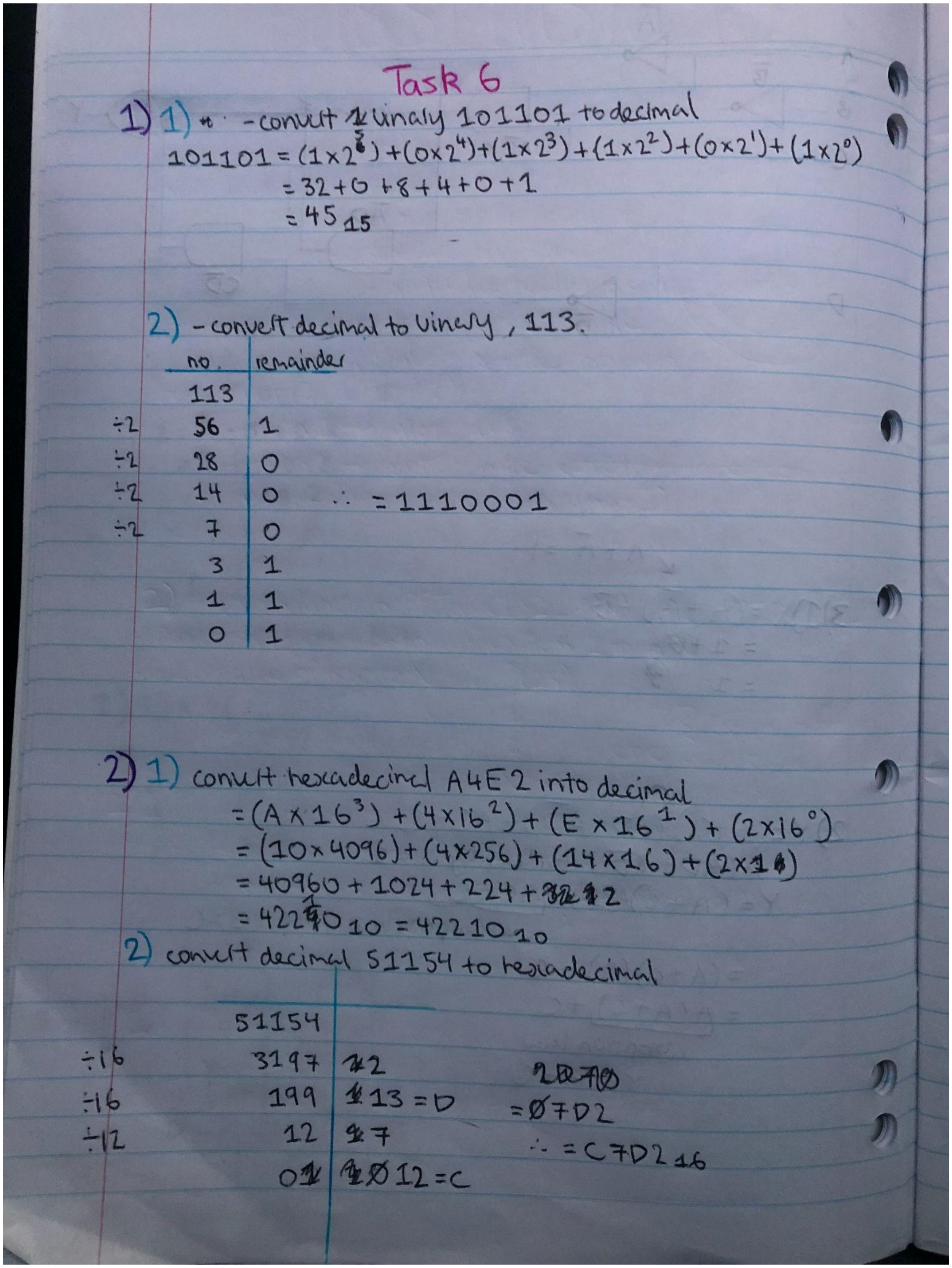
$= \underbrace{A(A+C)}_{\text{absoption law}} + C$

∴ $Y = A + C$

Task 6 :

Problem 1 :

Problem 2 :



Task 6

1) 1) - convert ~~1~~ binary 101101 to decimal

$101101 = (1\times2^{5})+(0\times2^{4})+(1\times2^{3})+(1\times2^{2})+(0\times2^{1})+(1\times2^{0})$

$= 32+0+8+4+0+1$

$= 45_{15}$

2) - convert decimal to binary , 113.

	no.	remainder
	113	
÷2	56	1
÷2	28	0
÷2	14	0
÷2	7	0
	3	1
	1	1
	0	1

∴ = 1110001

2) 1) convert hexadecimal A4E2 into decimal

$= (A\times16^{3})+(4\times16^{2})+(E\times16^{1})+(2\times16^{0})$

$= (10\times4096)+(4\times256)+(14\times16)+(2\times1)$

$= 40960+1024+224+2$

$= 42210_{10} = 42210_{10}$

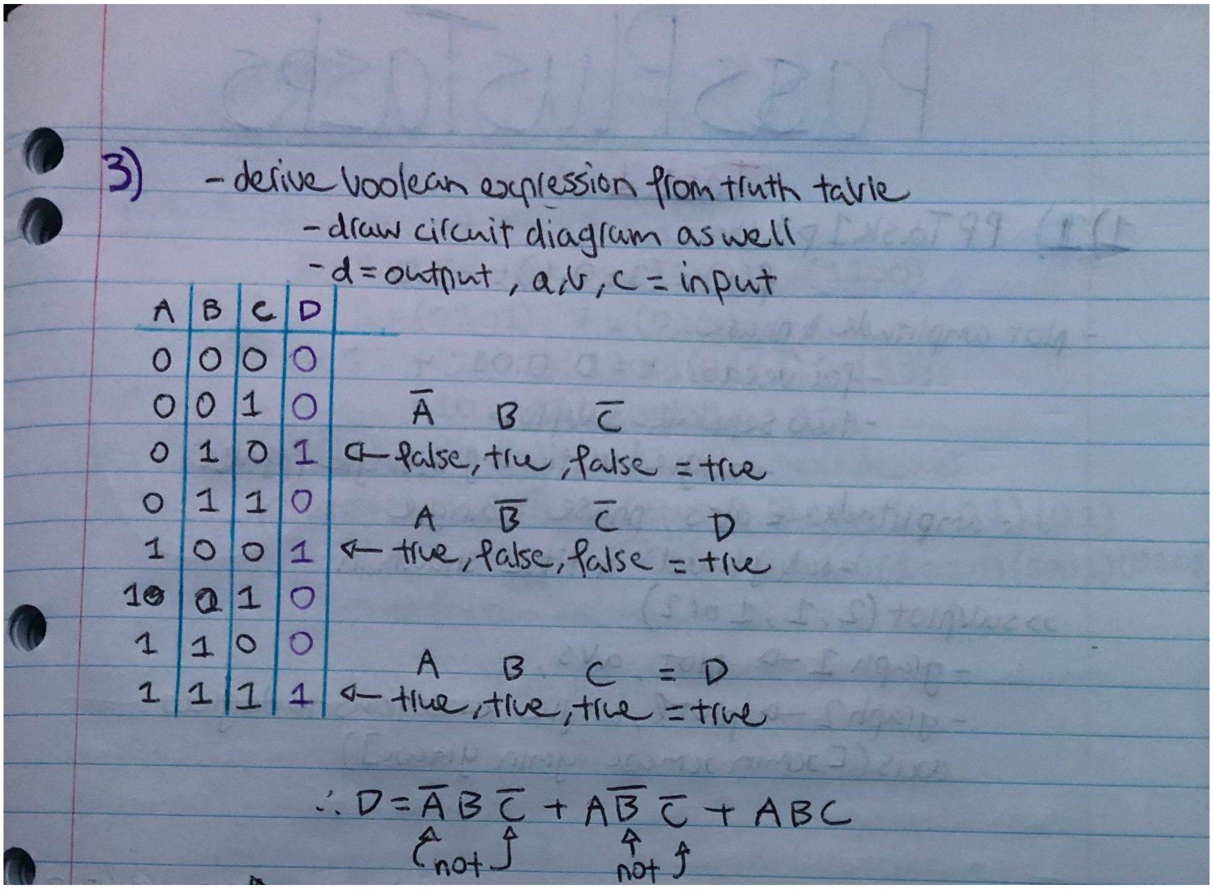
2) convert decimal 51154 to hexadecimal

	51154	
÷16	3197	2
÷16	199	13 = D
÷12	12	7
	0	12 = C

$= 07D2$

∴ $= C7D2_{16}$

Problem 3 :



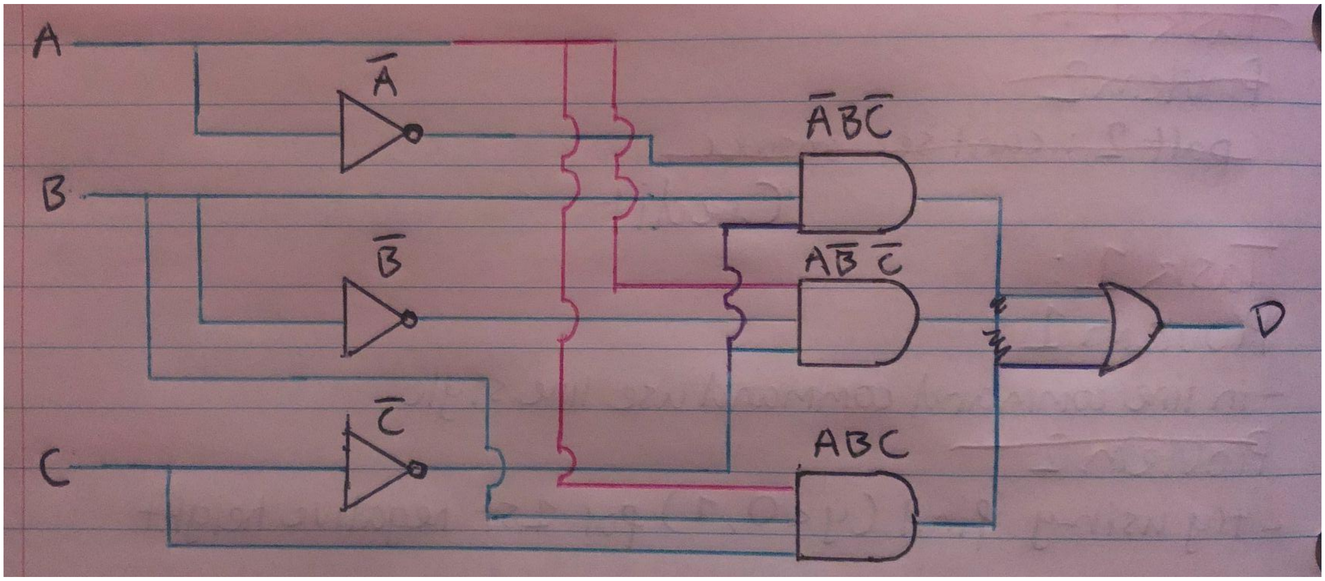
3) - derive boolean expression from truth table

- draw circuit diagram as well
- d = output , a,b,c = input

A	B	C	D	
0	0	0	0	
0	0	1	0	$\bar{A}$ B $\bar{C}$
0	1	0	1	← false, true, false = true
0	1	1	0	A $\bar{B}$ $\bar{C}$ D
1	0	0	1	← true, false, false = true
1	0	1	0	
1	1	0	0	A B C = D
1	1	1	1	← true, true, true = true

$$\therefore D = \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

not   not

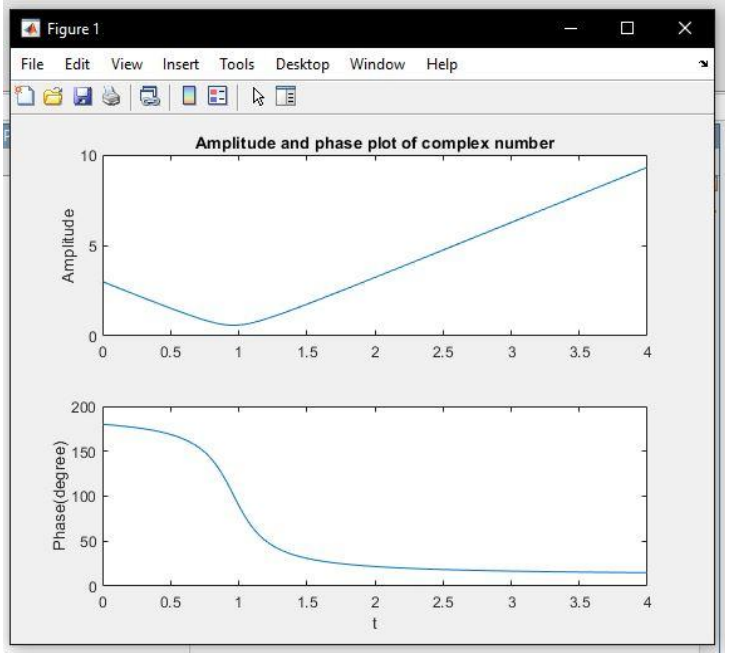


Pass Plus :

Task 1 :

Problem 1 :

```
t=0:0.01:4;
y_1=(3+0.6*i)*t-3;

subplot(2,1,1);
plot(t,abs(y_1));
axis([0 4 0 10])
ylabel('Amplitude');
title('Amplitude and phase plot of complex number');

subplot(2,1,2);
plot(t,angle(y_1)*180/pi);
axis([0 4 0 200]);
ylabel('Phase(degree)');
xlabel('t');
```



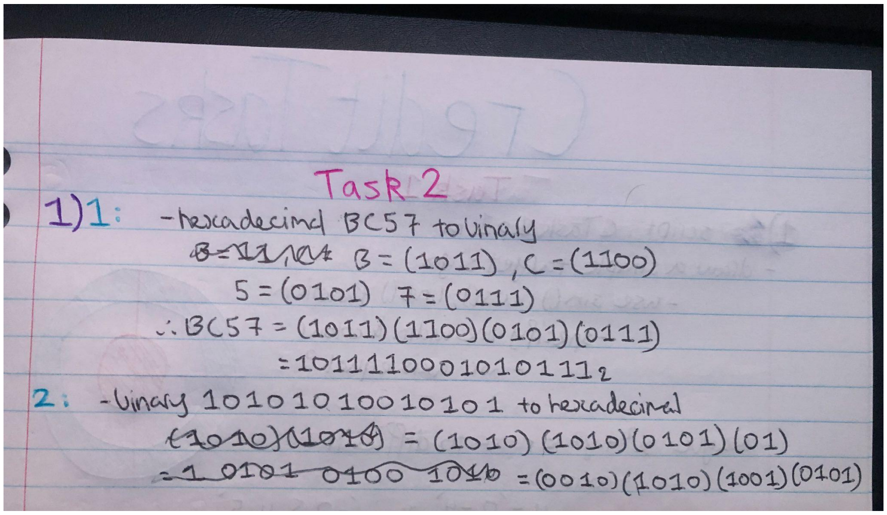
Problem 2 :

```
function PPTask1p2_f(N)

for k = 1:1:N
   r1 = rem(k,3); %if k is divisible by 3 or 5
   r2 = rem(k,5);
      if r2 == 0 && r1 == 0 %if remainder is equal to 0
         s1 = num2str(k); %turns into string so can be displayed
         disp([s1, ' is divisible by 3 & 5'])
      elseif r1 == 0
         s1 = num2str(k);

         disp([s1, ' is divisible by 3'])
      elseif r2 == 0
         s1 = num2str(k);
         disp([s1, ' is divisible by 5'])
      else
         s1 = num2str(k);
         disp([s1, ' is NOT divisible by 3 or 5'])
      end
end
end
```

```
>> PPTask1p2_f(25)
1 is NOT divisible by 3 or 5
2 is NOT divisible by 3 or 5
3 is divisible by 3
4 is NOT divisible by 3 or 5
5 is divisible by 5
6 is divisible by 3
7 is NOT divisible by 3 or 5
8 is NOT divisible by 3 or 5
9 is divisible by 3
10 is divisible by 5
11 is NOT divisible by 3 or 5
12 is divisible by 3
13 is NOT divisible by 3 or 5
14 is NOT divisible by 3 or 5
15 is divisible by 3 & 5
16 is NOT divisible by 3 or 5
17 is NOT divisible by 3 or 5
18 is divisible by 3
19 is NOT divisible by 3 or 5
20 is divisible by 5
21 is divisible by 3
22 is NOT divisible by 3 or 5
23 is NOT divisible by 3 or 5
24 is divisible by 3
25 is divisible by 5
```

Task 2 :

Problem 1 :



Task 2

1)1: - hexadecimal BC57 to binary

~~B = 11,101~~ B = (1011) , C = (1100)

5 = (0101) 7 = (0111)

∴ BC57 = (1011)(1100)(0101)(0111)

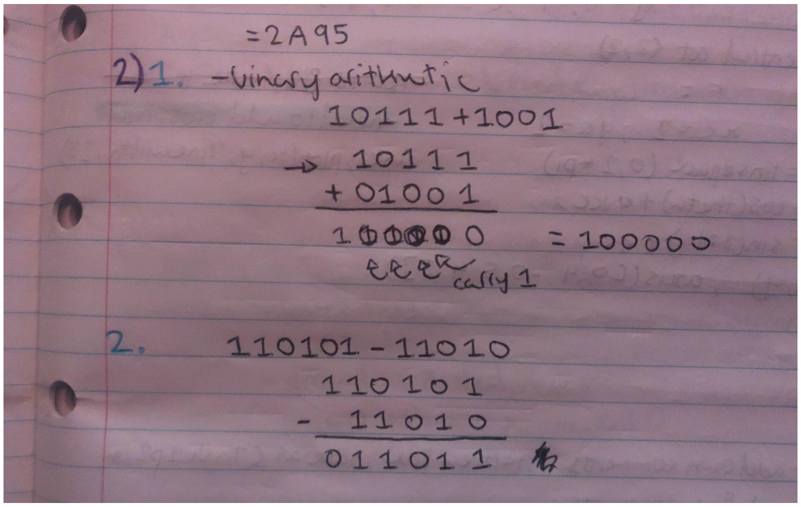
$= 1011110001010111_2$

2: - binary 1010101001010101 to hexadecimal

~~(1010)(1010)~~ = (1010) (1010)(0101)(01)

~~= 1 0101 0100 1010~~ = (0010)(1010)(1001)(0101)

Problem 2 :



=2A95

2)1. -binary arithmetic

10111+1001

→ 10111
+01001

100000 = 100000

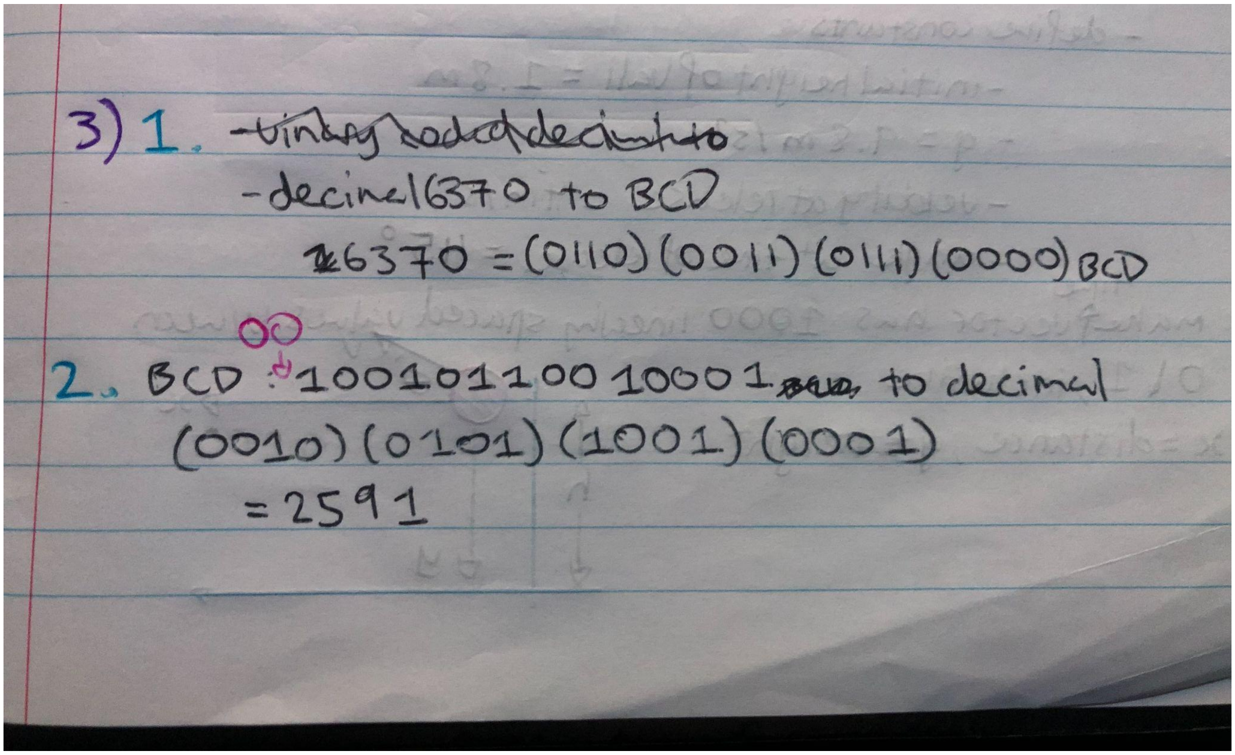
carry 1

2. 110101 - 11010

110101
- 11010

011011

Problem 3 :



3) 1. ~~-binary coded decimal to~~

-decimal 6370 to BCD

~~1~~6370 =(0110)(0011)(0111)(0000)BCD

00

2. BCD : 100101100 10001 ~~BCD~~ to decimal

(0010)(0101)(1001)(0001)

=2591

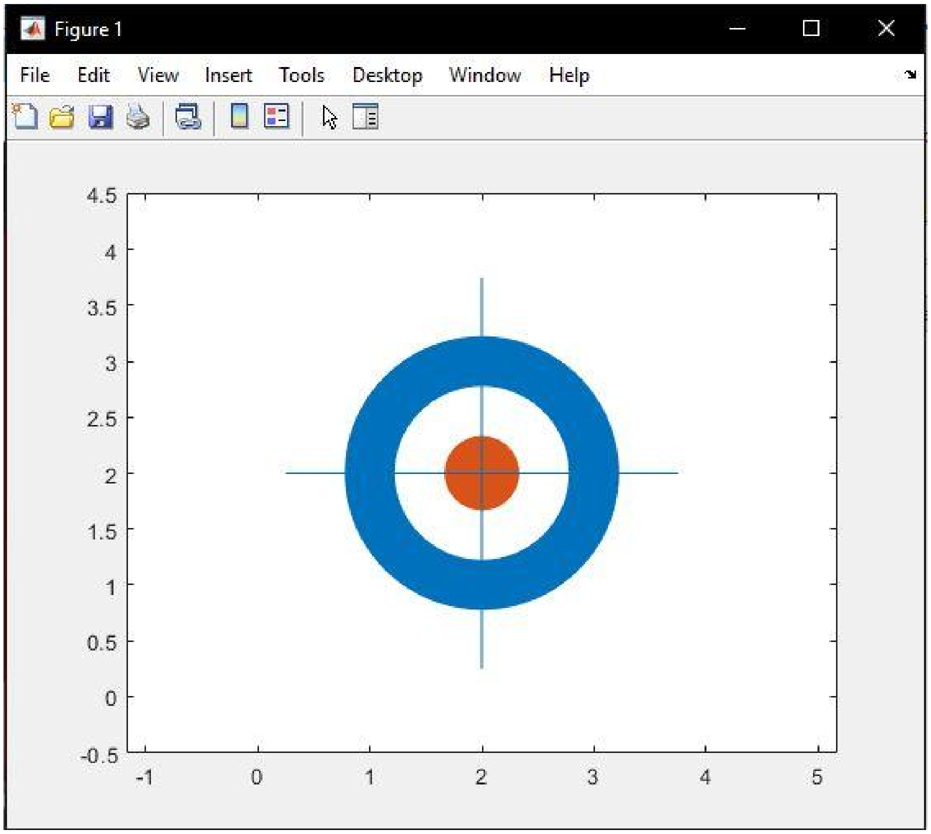
Joshua Lillington-Moore (103666887)

Credit :

Task 1 :

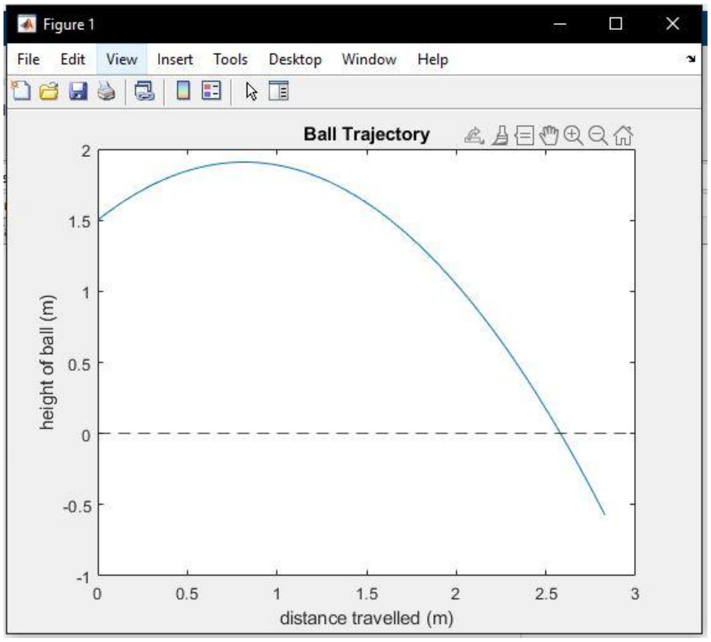
Problem 1 :

```
r = 1;
xc = 2;
yc = 2;

theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y,'linewidth',23)
axis ([0 4 -0.5 4.5])
axis equal

r = 0.1;
xc = 2;
yc = 2;

theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
hold on;
plot(x,y,'linewidth',24)
hold off;
axis ([0 4 -0.5 4.5])
axis equal

line([0.25 3.75],[2 2])
line([2 2],[0.25 3.75])
```



Problem 2 :

```
h=1.5;
g=9.8;
v=4;
angle=45;
t= 0:0.001:1;
x=v*cos(angle*(pi/180))*t;
y=h+v * sin(angle*(pi/180))* t-0.5*g*t.^2;
i=find(y<0,1);
xi = x(i);
```



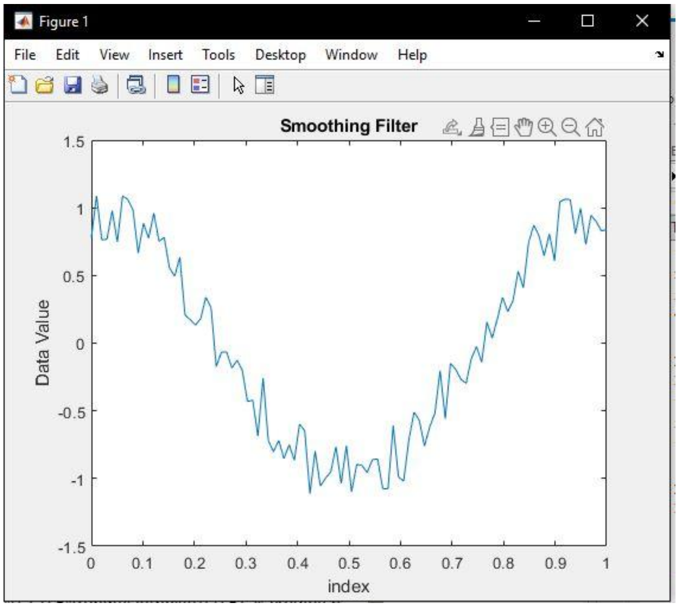
```
fprintf('The ball hits the ground at distance of %f metres \n',xi)
figure(1);
plot(x,y);
xlabel('distance travelled (m)');
ylabel('height of ball (m)');
title('Ball Trajectory');
hold on;
yline(0,'--k');
```

```
>> CTask1p2
The ball hits the ground at distance of 2.582354 metres
```

Task 2 :

Problem 1 :

```
smoothed = CTask2p1_f(x, width)

t = linspace(0,1,100);
noise = rand(1,length(t));
x = cos(2*pi*t) + 0.5*(rand(size(noise))-0.5); %creates noisy data set
width = 5;
figure
plot(t,x);
hold on;
title('Smoothing Filter')
xlabel('index')
ylabel('Data Value')
function smoothed
plot (t, smoothed, 'r')
hold on;
if rem(width,2)==0 %rem returns the remainder of width/2
   width=width + 1 %make sure width isn't positive
end
end
```



Problem 2 :