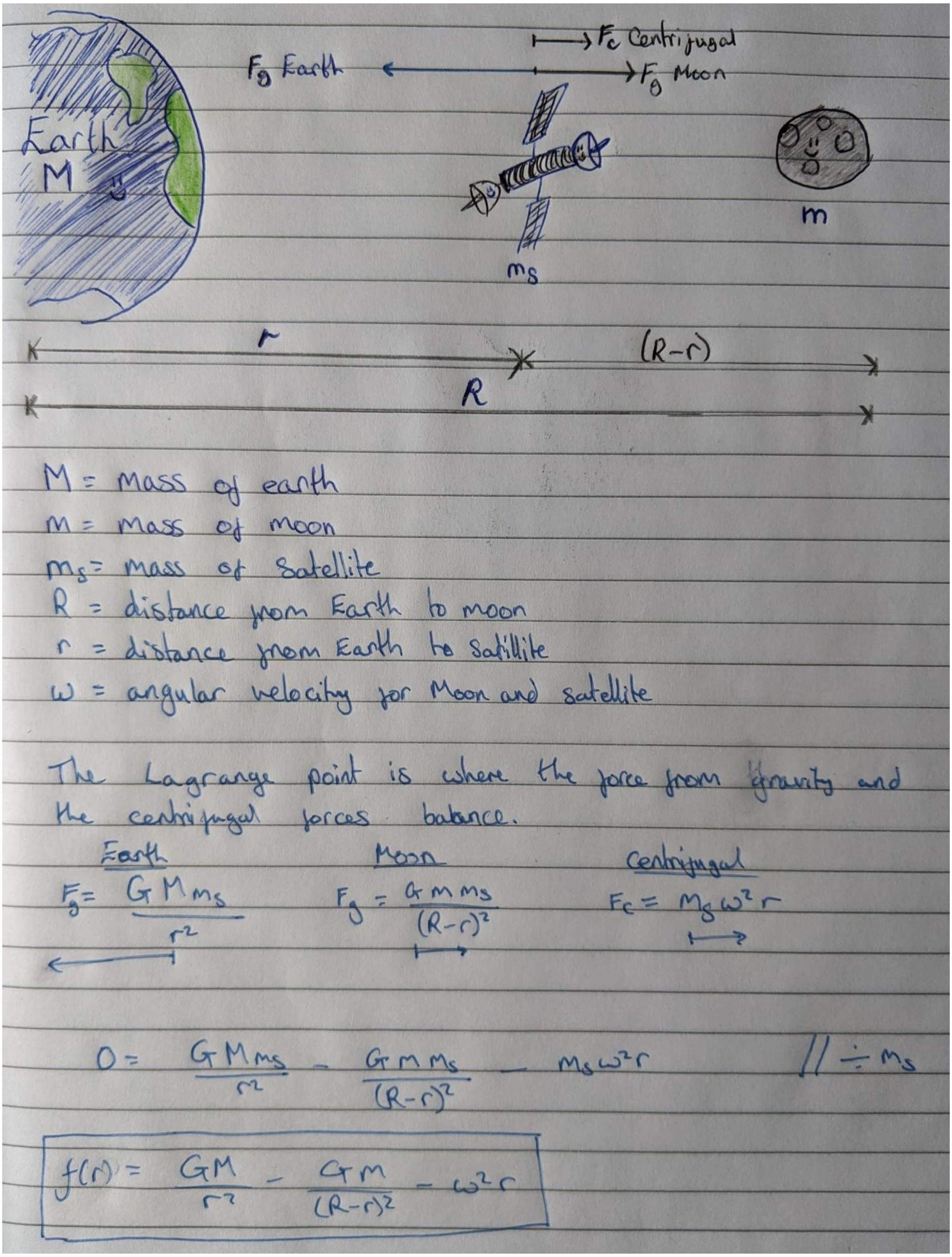
PH20018 Programming Skills: Coursework 1

1. Basic elements of C syntax

(a)



F_c Centrifugal
F_g Earth
F_g Moon
Earth
M
m_s
m
r
(R−r)
R

M = Mass of earth

m = mass of moon

m_s = mass of Satellite

R = distance from Earth to moon

r = distance from Earth to satillite

ω = angular velocity for Moon and satellite

The Lagrange point is where the force from gravity and the centrifugal forces balance.

Earth $\quad F_g = \frac{GMm_s}{r^2}$

Moon $\quad F_g = \frac{Gmm_s}{(R-r)^2}$

Centrifugal $\quad F_c = m_s\omega^2 r$

$$0 = \frac{GMm_s}{r^2} - \frac{Gmm_s}{(R-r)^2} - m_s\omega^2 r \qquad // \div m_s$$

$$\boxed{f(r) = \frac{GM}{r^2} - \frac{Gm}{(R-r)^2} - \omega^2 r}$$

✓

should not be handwritten

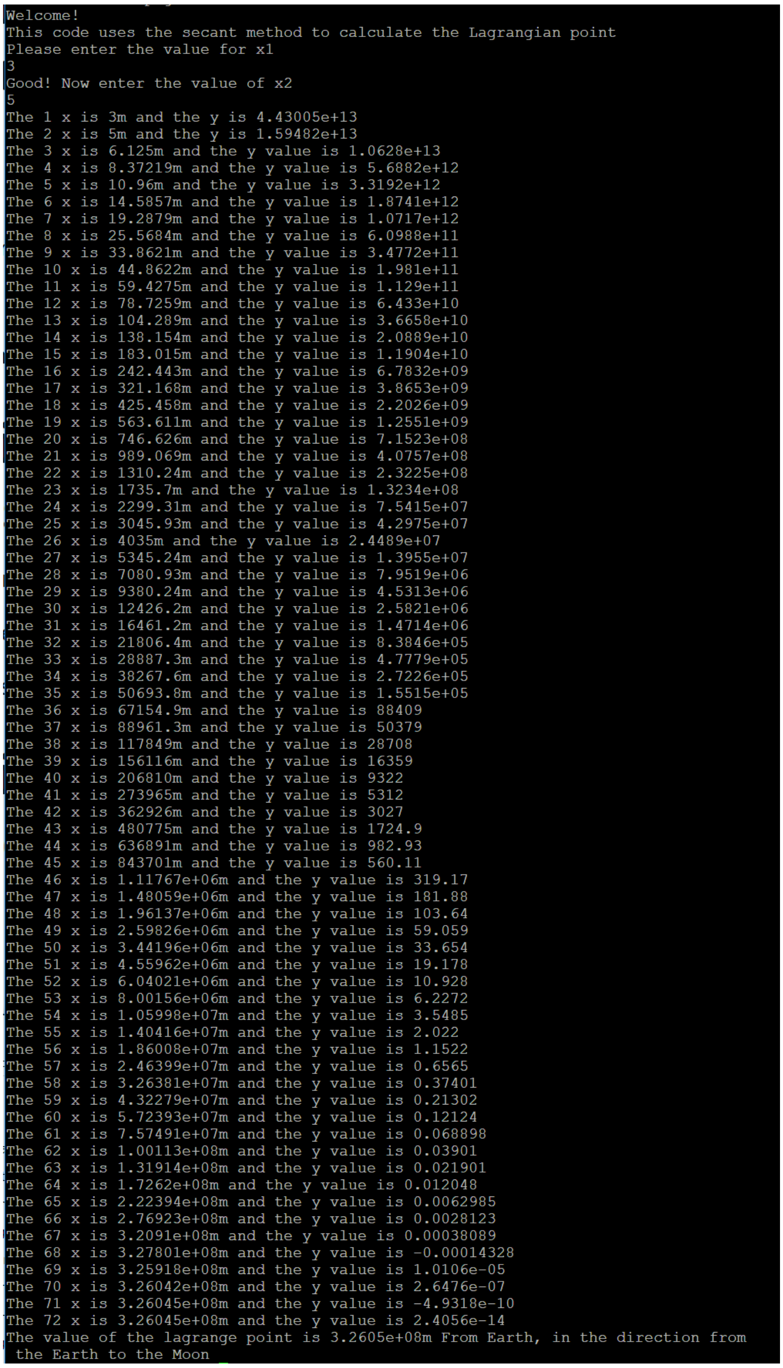
(b)

```
f(r) {a = $GM/r^2 - Gm/(R-r)^2 - w^2r$ //this is the function made 1(a)
      return a
      }

RoundTo5SigFig(Number){
      N = Log10(Number)
      N = floor(N)
      Number = Number* (10^(5-N))
      Number = round(Number)
      Number = Number * (10^(-5+N))
      Return Number
      }

MAIN {
INPUT r1 which is first value of r        how to choose these?
INPUT r2 which is second value of r
DO
      COMPUTE r3 using r3=r2-f(r2) * [(r2-r1)/(f(r2)-f(r1)]  //secant
                                                             method
      SET r2=r1
r1 = r2 and r2 = r3 ?
      SET r3=r2//MUST be in this order otherwise r2 will be deleted.
      UNTIL (RoundTo5SigFig(r1)==RoundTo5SigFig(r2))
// use round because there could be some small differences between
//r1 and r2 and it wont stop even though we only want answer to 5
//sig figs.
OUTPUT r3
}
```

(c)

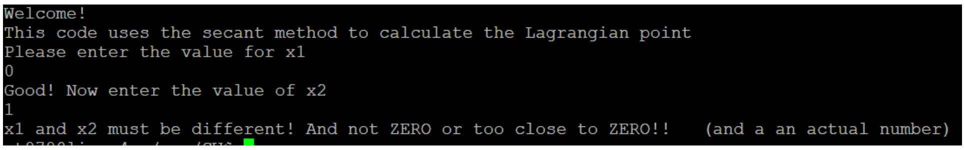
When x1 is 3 and x2 if 5 the correct answer appears. 3.2605x10^8m ✓

the used c-code should appear also in the report



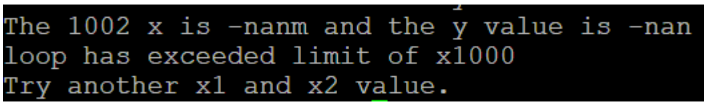
```
Welcome!
This code uses the secant method to calculate the Lagrangian point
Please enter the value for x1
3
Good! Now enter the value of x2
5
The 1 x is 3m and the y is 4.43005e+13
The 2 x is 5m and the y is 1.59482e+13
The 3 x is 6.125m and the y value is 1.0628e+13
The 4 x is 8.37219m and the y value is 5.6882e+12
The 5 x is 10.96m and the y value is 3.3192e+12
The 6 x is 14.5857m and the y value is 1.8741e+12
The 7 x is 19.2879m and the y value is 1.0717e+12
The 8 x is 25.5684m and the y value is 6.0988e+11
The 9 x is 33.8621m and the y value is 3.4772e+11
The 10 x is 44.8622m and the y value is 1.981e+11
The 11 x is 59.4275m and the y value is 1.129e+11
The 12 x is 78.7259m and the y value is 6.433e+10
The 13 x is 104.289m and the y value is 3.6658e+10
The 14 x is 138.154m and the y value is 2.0889e+10
The 15 x is 183.015m and the y value is 1.1904e+10
The 16 x is 242.443m and the y value is 6.7832e+09
The 17 x is 321.168m and the y value is 3.8653e+09
The 18 x is 425.458m and the y value is 2.2026e+09
The 19 x is 563.611m and the y value is 1.2551e+09
The 20 x is 746.626m and the y value is 7.1523e+08
The 21 x is 989.069m and the y value is 4.0757e+08
The 22 x is 1310.24m and the y value is 2.3225e+08
The 23 x is 1735.7m and the y value is 1.3234e+08
The 24 x is 2299.31m and the y value is 7.5415e+07
The 25 x is 3045.93m and the y value is 4.2975e+07
The 26 x is 4035m and the y value is 2.4489e+07
The 27 x is 5345.24m and the y value is 1.3955e+07
The 28 x is 7080.93m and the y value is 7.9519e+06
The 29 x is 9380.24m and the y value is 4.5313e+06
The 30 x is 12426.2m and the y value is 2.5821e+06
The 31 x is 16461.2m and the y value is 1.4714e+06
The 32 x is 21806.4m and the y value is 8.3846e+05
The 33 x is 28887.3m and the y value is 4.7779e+05
The 34 x is 38267.6m and the y value is 2.7226e+05
The 35 x is 50693.8m and the y value is 1.5515e+05
The 36 x is 67154.9m and the y value is 88409
The 37 x is 88961.3m and the y value is 50379
The 38 x is 117849m and the y value is 28708
The 39 x is 156116m and the y value is 16359
The 40 x is 206810m and the y value is 9322
The 41 x is 273965m and the y value is 5312
The 42 x is 362926m and the y value is 3027
The 43 x is 480775m and the y value is 1724.9
The 44 x is 636891m and the y value is 982.93
The 45 x is 843701m and the y value is 560.11
The 46 x is 1.11767e+06m and the y value is 319.17
The 47 x is 1.48059e+06m and the y value is 181.88
The 48 x is 1.96137e+06m and the y value is 103.64
The 49 x is 2.59826e+06m and the y value is 59.059
The 50 x is 3.44196e+06m and the y value is 33.654
The 51 x is 4.55962e+06m and the y value is 19.178
The 52 x is 6.04021e+06m and the y value is 10.928
The 53 x is 8.00156e+06m and the y value is 6.2272
The 54 x is 1.05998e+07m and the y value is 3.5485
The 55 x is 1.40416e+07m and the y value is 2.022
The 56 x is 1.86008e+07m and the y value is 1.1522
The 57 x is 2.46399e+07m and the y value is 0.6565
The 58 x is 3.26381e+07m and the y value is 0.37401
The 59 x is 4.32279e+07m and the y value is 0.21302
The 60 x is 5.72393e+07m and the y value is 0.12124
The 61 x is 7.57491e+07m and the y value is 0.068898
The 62 x is 1.00113e+08m and the y value is 0.03901
The 63 x is 1.31914e+08m and the y value is 0.021901
The 64 x is 1.7262e+08m and the y value is 0.012048
The 65 x is 2.22394e+08m and the y value is 0.0062985
The 66 x is 2.76923e+08m and the y value is 0.0028123
The 67 x is 3.2091e+08m and the y value is 0.00038089
The 68 x is 3.27801e+08m and the y value is -0.00014328
The 69 x is 3.25918e+08m and the y value is 1.0106e-05
The 70 x is 3.26042e+08m and the y value is 2.6476e-07
The 71 x is 3.26045e+08m and the y value is -4.9318e-10
The 72 x is 3.26045e+08m and the y value is 2.4056e-14
The value of the lagrange point is 3.2605e+08m From Earth, in the direction from
 the Earth to the Moon
```

If 0 is entered there is an error or if the number is too close to 0.



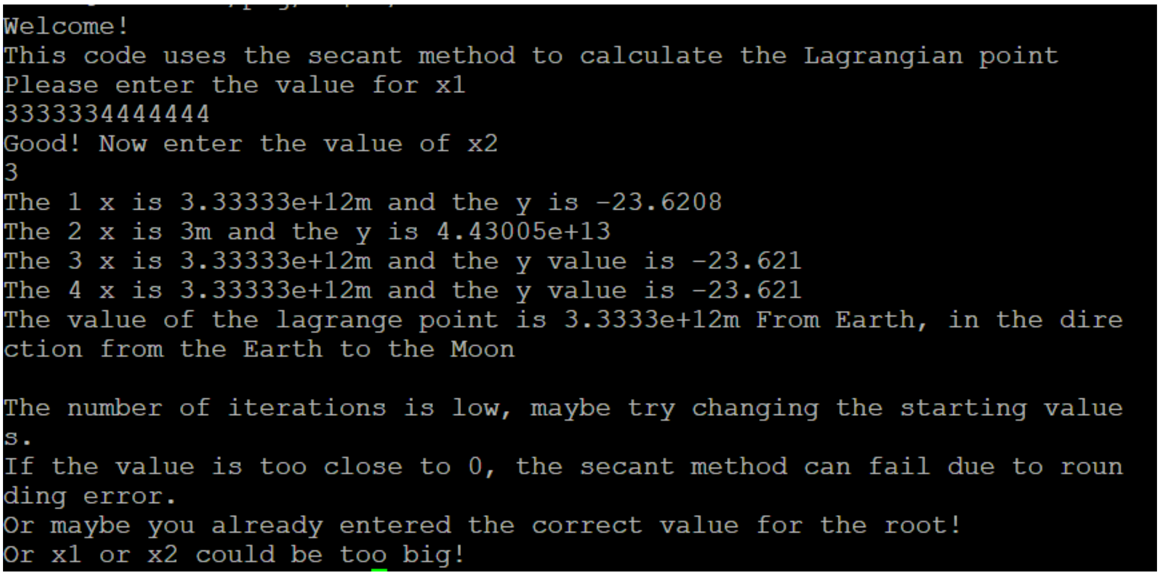
```
Welcome!
This code uses the secant method to calculate the Lagrangian point
Please enter the value for x1
0
Good! Now enter the value of x2
1
x1 and x2 must be different! And not ZERO or too close to ZERO!!   (and a an actual number)
```

Infinite loops cannot be avoided so a limit of 1000 is set



```
The 1002 x is -nanm and the y value is -nan
loop has exceeded limit of x1000
Try another x1 and x2 value.
```

Also because of rounding error, it could assume there is no change in x for large values. So it produces an error message in this case.



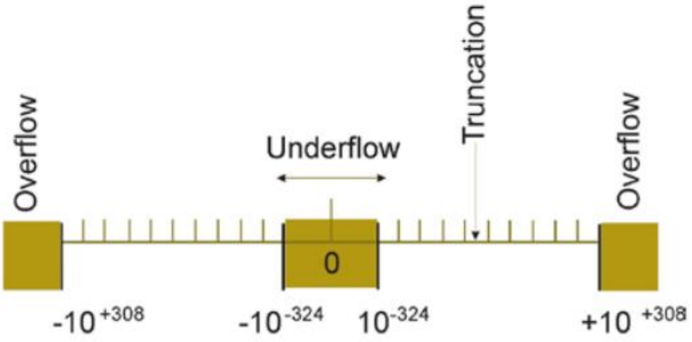
```
Welcome!
This code uses the secant method to calculate the Lagrangian point
Please enter the value for x1
3333334444444
Good! Now enter the value of x2
3
The 1 x is 3.33333e+12m and the y is -23.6208
The 2 x is 3m and the y is 4.43005e+13
The 3 x is 3.33333e+12m and the y value is -23.621
The 4 x is 3.33333e+12m and the y value is -23.621
The value of the lagrange point is 3.3333e+12m From Earth, in the dire
ction from the Earth to the Moon

The number of iterations is low, maybe try changing the starting value
s.
If the value is too close to 0, the secant method can fail due to roun
ding error.
Or maybe you already entered the correct value for the root!
Or x1 or x2 could be too big!
```

bit more discussion about the design of the program, could also have bit more discussion about the testing

17 points

2. Program design

(a) Discuss the limits on numerical precision imposed by the representation of real numbers in a computer



what defines the precision?

When numbers are too big(overflow), too negative(overflow), too close to zero(underflow), irrational(truncation) or are repeating decimal(truncation) then they cannot be truly stored in memory. This can lead to rounding error. An example of this is if you divide 10/3, and then multiply the answer by 3 you will get 9.9999. These errors can carry on and accumulate in long processes. Coders must be aware of these and take precaution when needed. There are ways of helping with rounding errors. One way is by using larger variable types such as double float. However, they come with the drawback of slower programs. And they still experience the same issues.

(b) What conclusions about the nature of the numerical integration problem can one draw from inspection of the equation?

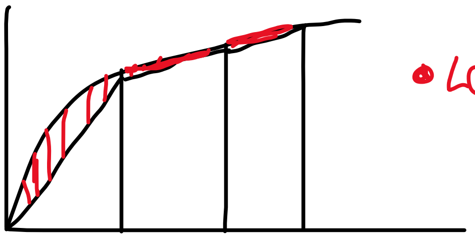
When 0 is entered into the equation, there is no "answer". 0 divided by 0 is not a valid. But in our case, we will have to assume the answer is 0. Sin(x)/sqrt(x) also goes above and below zero. Like a sign graph. This is a problem because the integral would also increase then decrease in value around a point. In this it is heading towards the $(pi/2)^{1/2}$. The larger the value the closer it gets to this value. This integral also has a period of 2 pi. However, to get close to the value, very large values would have to be used.

explain the assumption?

why this is a problem?

graph would be helpful

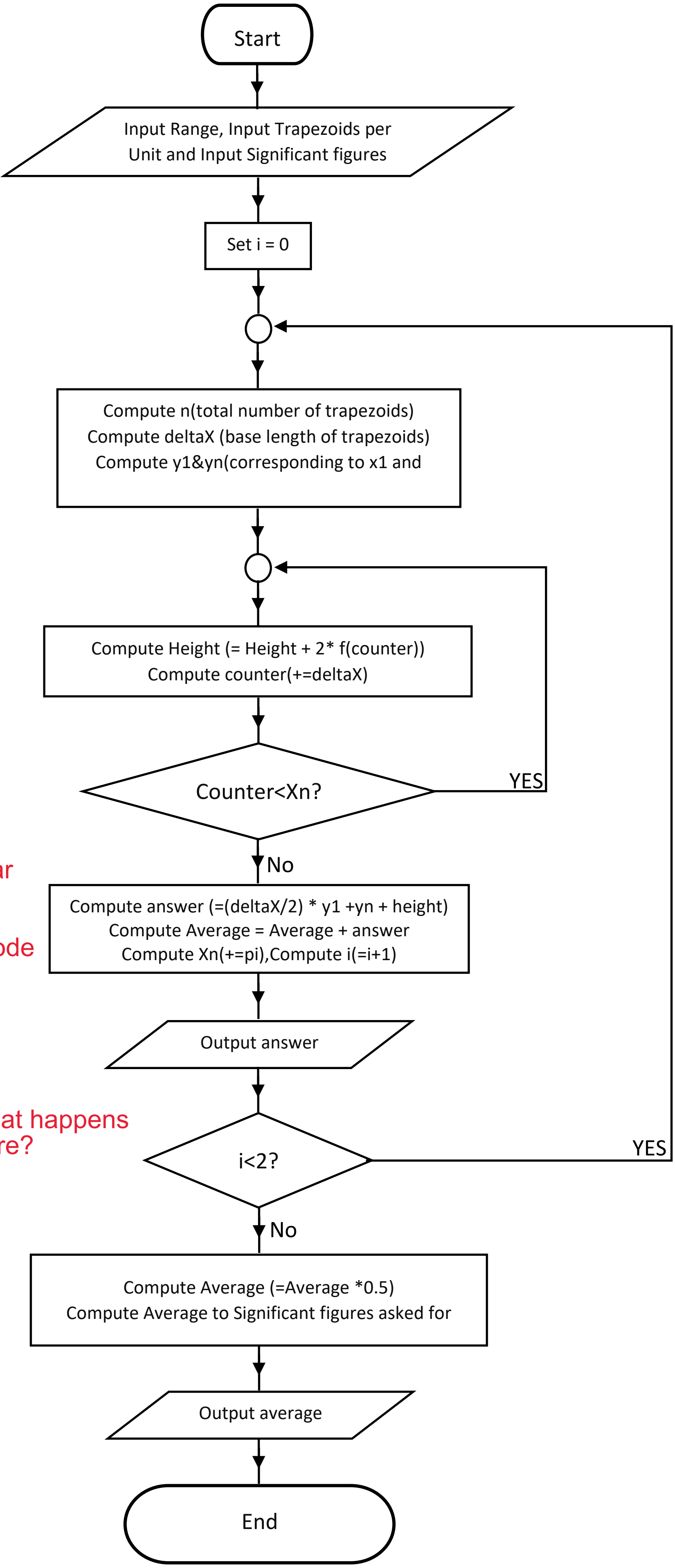
This can be shown by this example. say we integrate from 0 to 1000000. Sqrt(10000000) is 1000, so the sin(x)/sqrt(x) would still oscillate by 1/1000=0.001. Which, when you include the error of the trapezium method underestimating the area under the graph, will add up. ✓



• Lost

(c) Flow chart

could these be automated according to the required precision?


Start
Input Range, Input Trapezoids per Unit and Input Significant figures
Set i = 0
Compute n(total number of trapezoids)
Compute deltaX (base length of trapezoids)
Compute y1&yn(corresponding to x1 and
Compute Height (= Height + 2* f(counter))
Compute counter(+=deltaX)
Counter<Xn?
YES
No
Compute answer (=(deltaX/2) * y1 +yn + height)
Compute Average = Average + answer
Compute Xn(+=pi),Compute i(=i+1)
Output answer
i<2?
YES
No
Compute Average (=Average *0.5)
Compute Average to Significant figures asked for
Output average
End

this part is not very clear

bit like a mixture of a flowchart and pseudocode

what happens here?

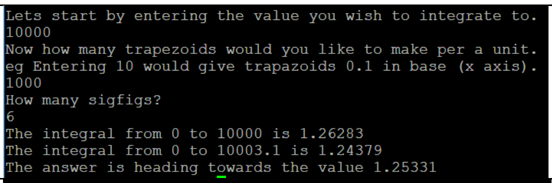
14 points

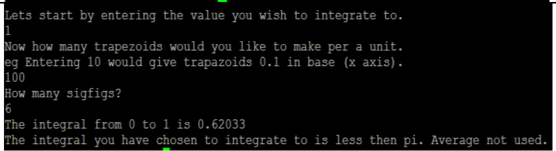
the used c-code should appear also in the report

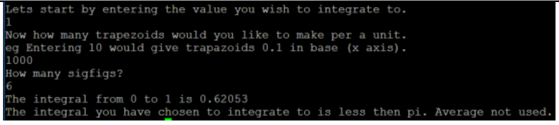
3. C programming and debugging

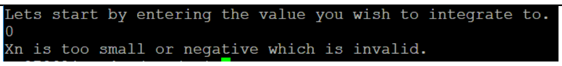
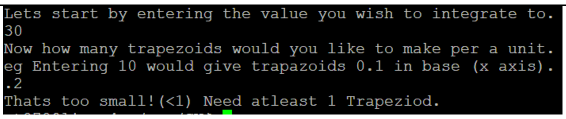
Integral of Sin(x)/sqrt(x) from 0 to inf is 1.253314

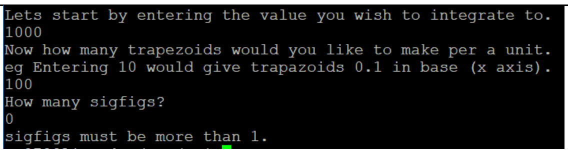
	Input	Output	screenshot	Notes

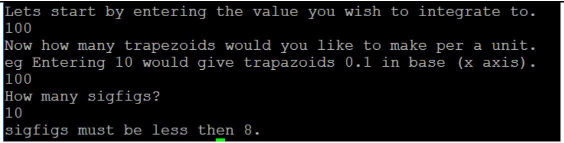
1.	Xn=10000
	No trapezoids per unit =1000
	Sigfigs=6
		1.25331	
			Lets start by entering the value you wish to integrate to.
			10000
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			1000
			How many sigfigs?
			6
			The integral from 0 to 10000 is 1.26283
			The integral from 0 to 10003.1 is 1.24379
			The answer is heading towards the value 1.25331
				The average is correct to 6 sigfigs. This is very good. However, 6 sigfigs is the max

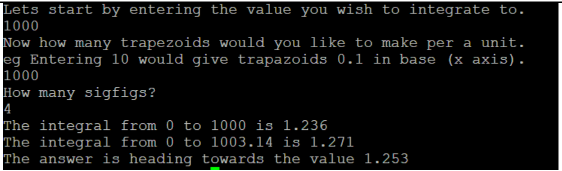
2.	Xn=1
	No trapezoids per unit =100
	Sigfigs=6
		0.62033	
			Lets start by entering the value you wish to integrate to.
			1
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			100
			How many sigfigs?
			6
			The integral from 0 to 1 is 0.62033
			The integral you have chosen to integrate to is less then pi. Average not used.
				This value is also very close to true value. The real value is 0.620537

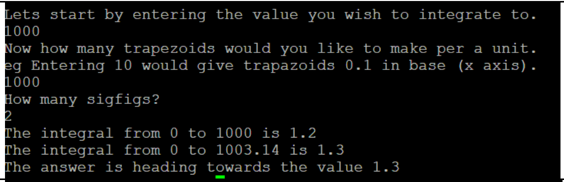
3.	Xn=1
	No trapezoids per unit =1000
	Sigfigs=6
		0.62053	
			Lets start by entering the value you wish to integrate to.
			1
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			1000
			How many sigfigs?
			6
			The integral from 0 to 1 is 0.62053
			The integral you have chosen to integrate to is less then pi. Average not used.
				This value is 0.00001 off.

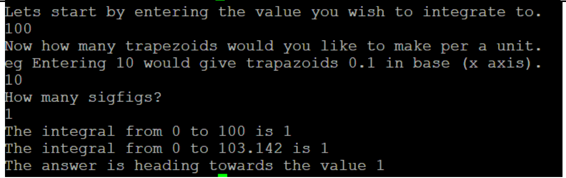
4.	Xn=0	Error message	 Lets start by entering the value you wish to integrate to. 0 Xn is too small or negative which is invalid.	If value you wish to integrate to is 0, error message is displayed.
5.	Xn=0 No trapezoids per unit =.2	Error message	 Lets start by entering the value you wish to integrate to. 30 Now how many trapezoids would you like to make per a unit. eg Entering 10 would give trapazoids 0.1 in base (x axis). .2 Thats too small!(<1) Need atleast 1 Trapeziod.	Error message when invalid trapezoid per unit entered

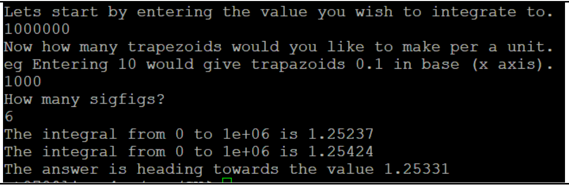
6.	Xn=100
	No trapezoids per unit =10
	Sigfigs=.2
		Error message	
			Lets start by entering the value you wish to integrate to.
			1000
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			100
			How many sigfigs?
			0
			sigfigs must be more than 1.
				Error message when invalid sig figs entered.

7.	Xn=100
	No trapezoids per unit =10
	Sigfigs=10
		Error message	
			Lets start by entering the value you wish to integrate to.
			100
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			100
			How many sigfigs?
			10
			sigfigs must be less then 8.
				Error message when too large of sigfig entered

8.	Xn=1000
	No trapezoids per unit =1000
	Sigfigs=4
		1.253	
			Lets start by entering the value you wish to integrate to.
			1000
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			1000
			How many sigfigs?
			4
			The integral from 0 to 1000 is 1.236
			The integral from 0 to 1003.14 is 1.271
			The answer is heading towards the value 1.253
				Correct answer. To 4 sigfigs.

9.	Xn=1000
	No trapezoids per unit =1000
	Sigfigs=2
		1.3	
			Lets start by entering the value you wish to integrate to.
			1000
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			1000
			How many sigfigs?
			2
			The integral from 0 to 1000 is 1.2
			The integral from 0 to 1003.14 is 1.3
			The answer is heading towards the value 1.3
				Correct answer

10.	Xn=100
	No trapezoids per unit =10
	Sigfigs=1
		1	
			Lets start by entering the value you wish to integrate to.
			100
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			10
			How many sigfigs?
			1
			The integral from 0 to 100 is 1
			The integral from 0 to 103.142 is 1
			The answer is heading towards the value 1
				Correct answer

11.	Xn=10000000
	No trapezoids per unit =1000
	Sigfigs=6
		1.25331	
			Lets start by entering the value you wish to integrate to.
			1000000
			Now how many trapezoids would you like to make per a unit.
			eg Entering 10 would give trapazoids 0.1 in base (x axis).
			1000
			How many sigfigs?
			6
			The integral from 0 to 1e+06 is 1.25237
			The integral from 0 to 1e+06 is 1.25424
			The answer is heading towards the value 1.25331
				Correct. Same answer as first test

The program outputs the answer to the correct significant figure. For the definite integral the answer is correct to 6 significant figures as shown by test 1. From above you can see it produces the correct results. When I was first making the code, I didn't take the average of 2 points. This made the code very slow. I tried to take the integral from 0 to 1000000, but even this was not very accurate as it would be off by 0.001. I changed the program to take 2 integrals that are pi apart. Then finds the average.

explain bit more

Taking the average gives the correct integral to 6 significant figures. This is a massive improvement. This can be seen in test 11. Where, with out the average, the answer would be 1.25237, and after the average it is 1.25331. You get the same results from test 1, except test 1 is instant and test 11 takes about minutes to complete. I chose to allow my user to pick significant figures to determine how accurate they want the answer. The answer is given to the correct significant figures as shown in the tests (8,9,10). There are also error messages in place for when the user enters values that do not make sense. E.g. the integral from 0 to 0.

good that you carried out some tests

bit more discussion about the design of the program

some automation would be useful, e.g. for the step size, since you should not expect that the user knows what type of values to input

32 points

total score 63