GNG1106 Winter 2018 - Assignment 5

Due: Mar 4, 11:30 pm

Instructions

This assignment is to be done INDIVIDUALLY. Use the following instructions to complete and submit this assignment.

- You will need to submit your assignment electronically to Brightspace. Prepare the following
  - An assignment report in a PDF file (this allows you to use your favorite editor to create the PDF file). For question 1, insert the programming model filled in as per the question instructions. You may fill in the programming model using drawing features of your editor or by hand on paper which is then scanned and inserted into your document. For Questions 2 and 3, for this assignment it is **NOT** necessary to insert in your assignment report the source code but do insert the output from running the program for all test cases. You must submit your source code files for question 2 and 3.
- Place all your files (PDF file and C source code files) in a directory A5_xxxxxxxx where xxxxxxxx is your student number.
- Zip your PDF document and the C source files in a zip file with the name A5_xxxxxx.zip where xxxxxx is your student number.
- Submit the zip file before the assignment deadline via Brightspace. In Brightspace, navigate to the Assignment page and click on "Click to submit Assignment 5" to reach the assignment 5 submission folder. You can also select the Assignment tab to see the Assignment folder pages. The Brightspace video "Assignments" (found in the page https://tlss.uottawa.ca/site/en/connection-to-the-lms#video-tutorials) provides details to help you submit the zip file.
- The questions are provided in both PDF and Word files. You may use the Word file (please remove the questions) to enter your answers in the document. An rtf file is also provided so that you may edit the file with a word processor other than Word. Do save your file as a PDF file for submission.

➢ Do start the assignment soon and do **not** wait until the last minute. You will be more efficient with a number of smaller efforts over a few weeks before the deadline than one large effort just before the deadline.

Marking Scheme (total 30 marks)

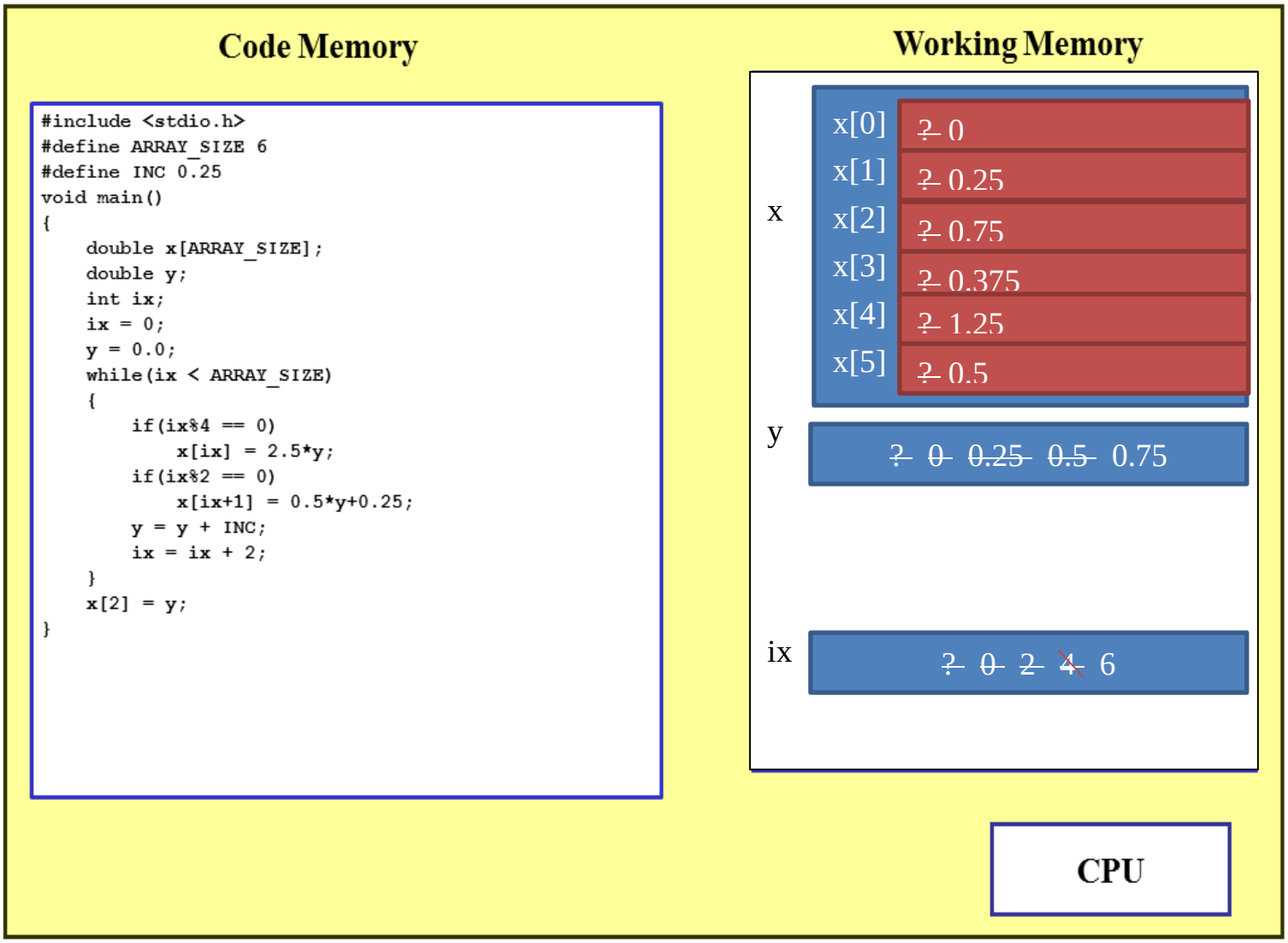
- Question 1: 5 marks
- Question 2: 10 marks
- Question 3: 15 marks

Question 1 (5 marks)

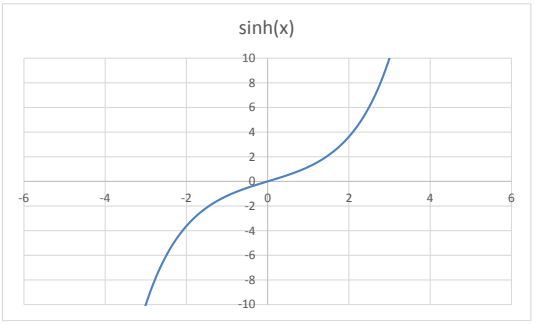
The following programming model contains in its code memory the indicated C program composed of a `main` function. You will be showing how the working memory is used during the execution of this function. Show how the given C program affects the contents of the working memory:

- Show how the array and variables occupy space in the working memory.
- Show how the variables and arrays are initialized and updated by the loop.
- Show the values are assigned the variable and elements in the array by the assignment instructions. Be sure to show all values that are assigned and replaced. Record successive assignments to variables/parameters as follows (the ? shows an unknown value):

varName	~~?~~, ~~2~~, ~~6~~, ~~4~~, 10



Question 2 (10 marks)



Ever wonder how the standard C math functions make their calculations. This question lets you explore how to compute hyperbolic sine of any value of *x*, sinh(*x*). The following infinite series gives the value of sinh(*x*):

$$\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots = \sum_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)!} \qquad \text{Equation 1}$$

Such series serves as a basis for calculating sinh(*x*) in a program (i.e. in a function). Of course a program has to limit the number of terms for computing the value (it is impossible to compute an infinite number of terms). Thus the above equation becomes:

$$\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots = \sum_{i=0}^{N} \frac{x^{2i+1}}{(2i+1)!} \qquad \text{Equation 2}$$

Develop software that meets the following requirements:

- Prompts the user for the value of *x* and the number of terms, n, in the series used to compute sinh(*x*).
  - Check that the number of terms given by the user is positive.
- Computes the value of sinh(*x*) using the above series and displays the results to the user.
- Prompts the user to quit the program with the message "Do you wish to quit (y/n)? ", and read in a character. Repeat the above steps if the user enters 'y' and stop the program if the user answers 'n'. Keep prompting the user if an invalid answer is entered.
- Print the message "Program terminated" when the program terminates.

Respect the following guidelines:

- DO NOT use arrays or structures in your answer.
- DO NOT use standard math functions in your answer.
- Have the **`main`** function call a separate function, `getInput`, to get input values from the user. Recall that you can pass addresses to simple variables to a function in a parameter so that the function can fill in the variable value.
- Have the **`main`** function call a separate function, `sinHyper`, to compute the value of sinh(*x*).
  - Given that the powers of *x* in the numerator and the factorial in the denominator will become very large in a very few terms, calculating the values of each term separately will quickly reach the limit of the computer.
  - Use a strategy where the value of a term is calculated from the previous term. If we number the terms using *i* starting at 0 (and the term t_0 = x), then the term t_i can be computed from the previous term t_{i-1} as follows:

$$t_i = \frac{x^2}{(2i+1)2i} t_{i-1}$$

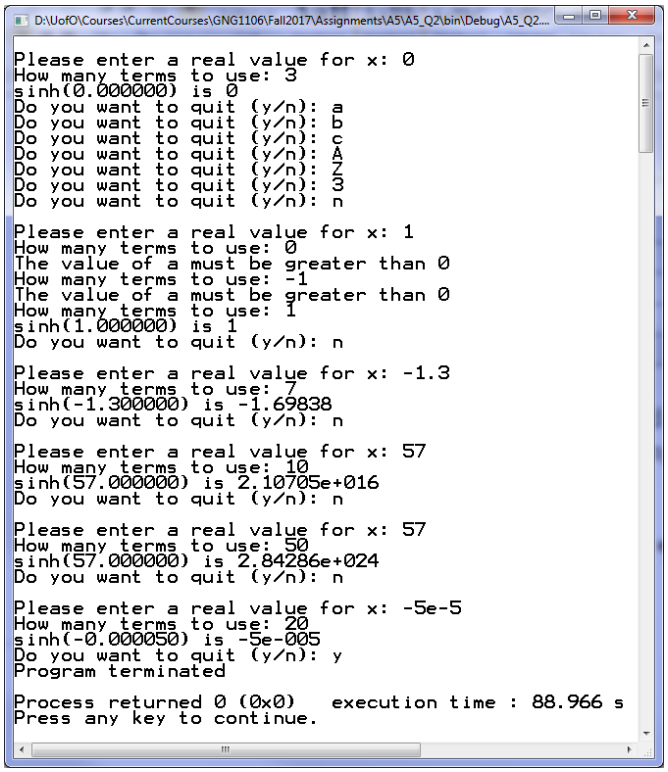
  - The above computation does not involve the same large numbers as the numerator/denominator of the terms in Equation 2.
  - If n is the number of terms used to calculate sinh(*x*), then *i* varies between 0 and *n*-1 (i.e. N = n-1 in Equation 2).
- A loop to control termination of the program is part of the **`main`** function.
  - Recall that `fflush` function should be called before reading in a character input from the user.

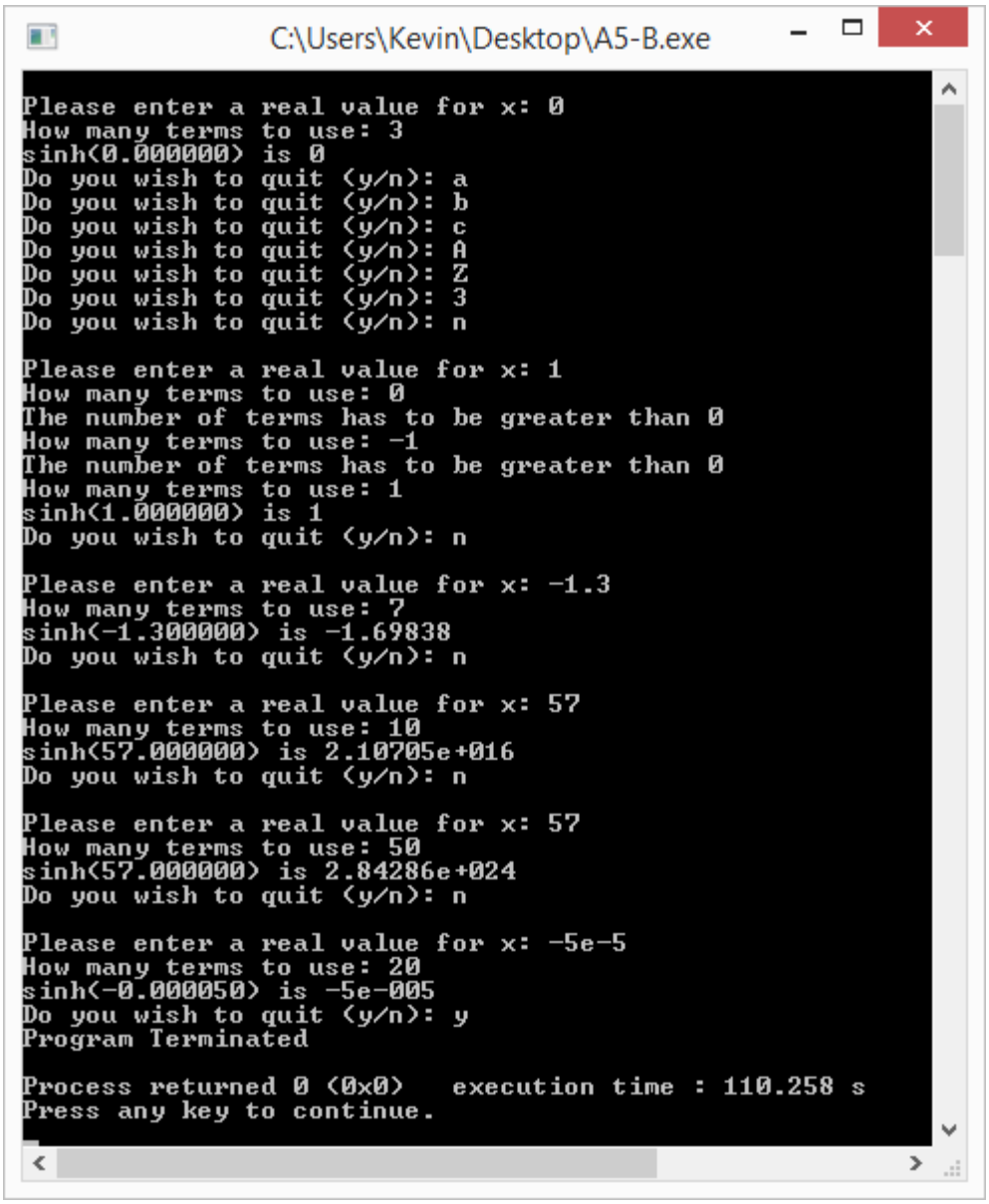
To answer this question, please provide:

1) The source code to your program in a C file (DO NOT insert the source code into your assignment file, PDF file).
2) Insert the output for the following test cases into your assignment document. Include testing of bad values for n and the answer when prompting to quit.

x	n (number of terms)	sinh(*x*)
0	3	0
1	1	1.175201194
-1.3	7	-1.698382437
57	10	2.84286E+24
57	50	2.84286E+24
-5.60E-05	30	-5.60E-05

Note that the values in the above table are calculated using Excel; the software may not give identical results, particularly for low value of n. The following shows a sample output that tests the above values as well as improper input.





Question 3 (15 marks)

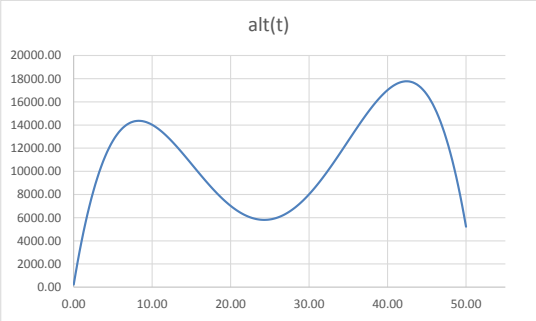
A weather balloon is used to collect data on temperature and pressure at various altitudes in the atmosphere. The balloon rises because the density of the helium in the balloon is lower than the density of the air that surrounds the balloon. As the balloon rises, the air around the balloon becomes less and less dense, and thus the rising of the balloon slows until it reaches a point of equilibrium During the day, the sun warms the helium in the balloon, which causes the helium to expand and become less dense; thus, the balloon rises to a higher altitude. During the night, the helium cools and becomes denser; thus, the balloon lowers to a lower altitude. The following day, the sun heats the helium again and the balloon rises to a new altitude.

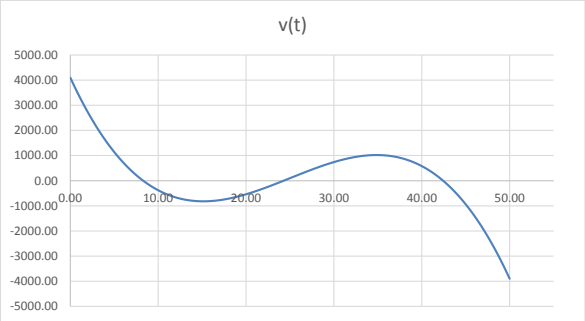
The change in the weather balloon altitude as a function of time can be approximated with a polynomial equation. Assume that the change in altitude in meters during the first 48 hours following the release of the balloon is represented by the equation:

$$alt(t) = -0.12t^4 + 12t^3 - 380t^2 + 4100t + 220$$

where the unit of time *t* is hour. The corresponding velocity in meters per second of the balloon is represented by the equation

$$v(t) = -0.48t^3 + 36t^2 - 760t + 4100\,.$$





Develop a program that requests from the user a range of time for t to calculate the change in altitude and velocity of the weather balloon. Twenty-five values of time/altitude/velocity over this range are displayed in a table. Respect the following guidelines during your development:

a) Declare a structure variable in the **main** function which contains:
   - The start time and end time of the period to calculate altitude and speed values,
   - A computer array for storing time values,
   - A computer array for storing altitude values,
   - A computer array for storing velocity values,
   - The values at the same index of the three arrays give the altitude and velocity at the time value.
   - The arrays will contain 25 elements. Use a symbolic constant to represent this value.

b) Obtain from the user the following data:
   - The start and end time of the of the period to calculate altitude and speed values,
   - Verify that the two time values are between 0.0 and 48.0. Also verify that the start time proceeds the end time. Repeat the request from the user for the times values until they are both valid.
   - Use a separate function (called by **main**) to get the user data. The function should receive the address of the variable structure declared in **main** within a pointer parameter to store the two time values.

c) Use a function to calculate all time/altitude/velocity points to fill the three arrays in the structure variable. The function shall also receive the address to the structure variable declared in **main** within a pointer parameter in order to update the arrays of the structure variable.

d) The **main** function displays in a table how the altitude and velocity varies as a function of time over the given period. Use the following format.

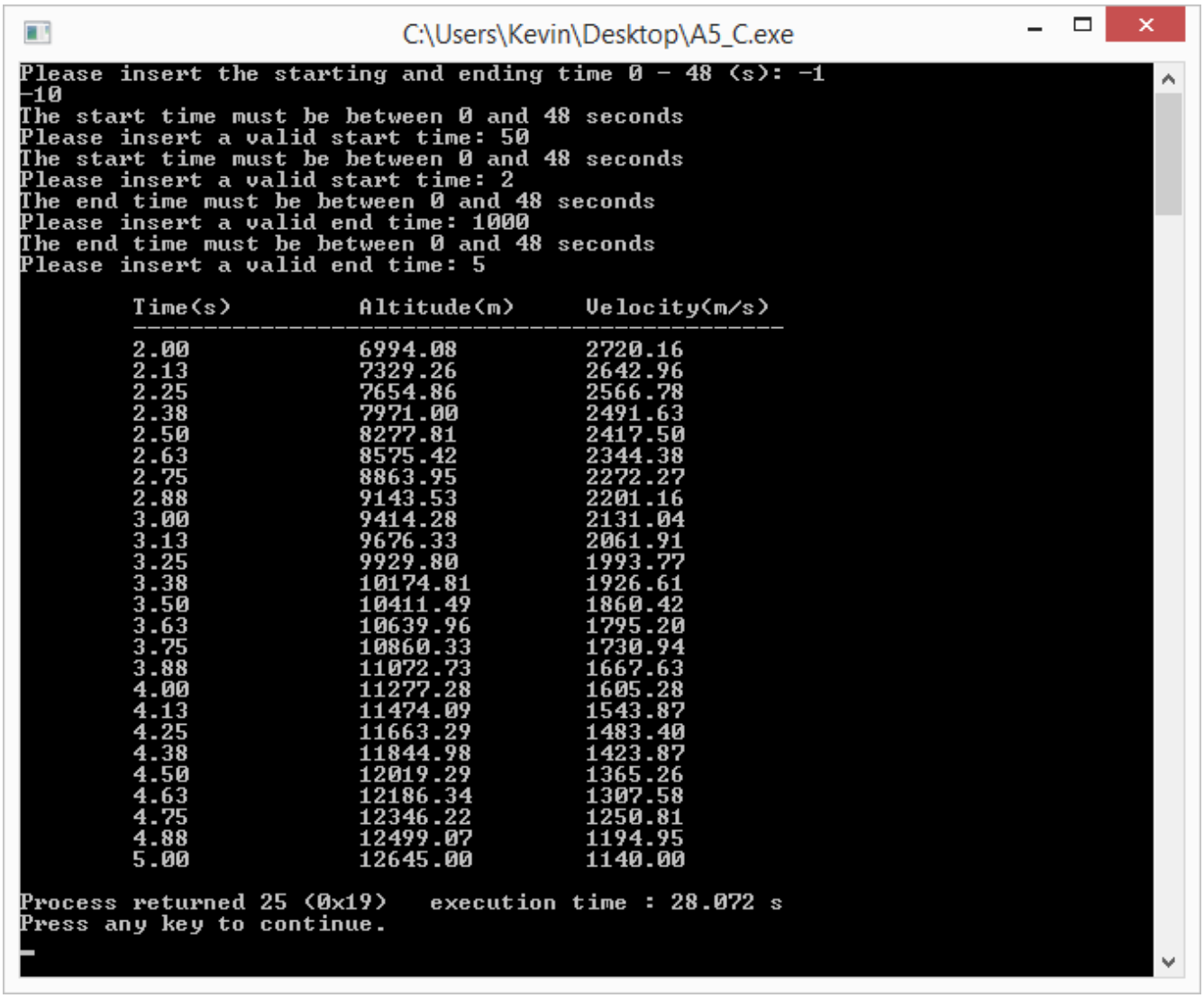
```
    Time      Altitude    Velocity
-----------------------------------
   35.00      12645.00     1020.00
   35.42      13069.30     1015.80
   35.83      13490.78     1006.39
   36.25      13907.21      991.56
   36.67      14316.30      971.11
   37.08      14715.66      944.83
                .            .
                .            .
                .            .
```

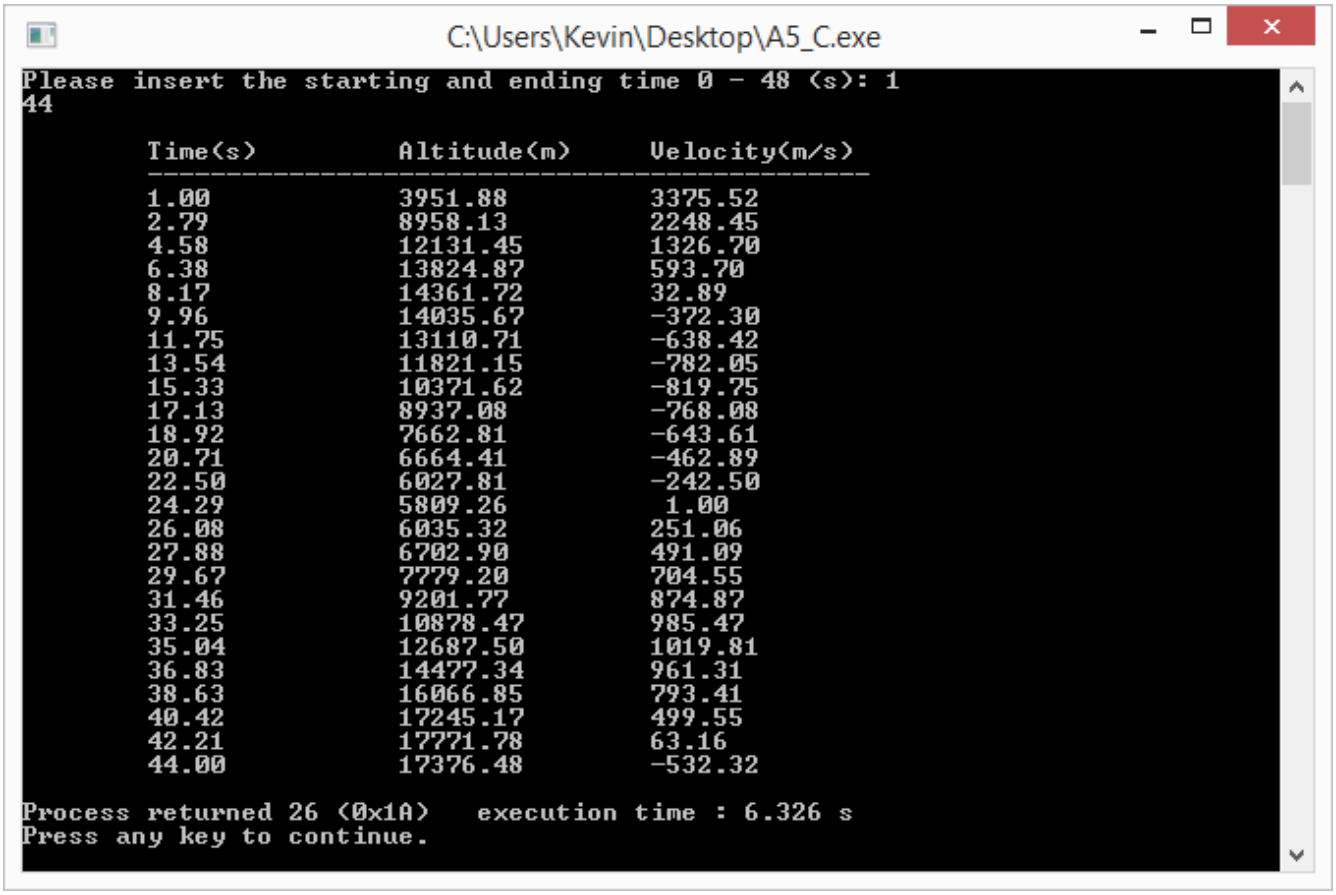
To answer this question, please provide:

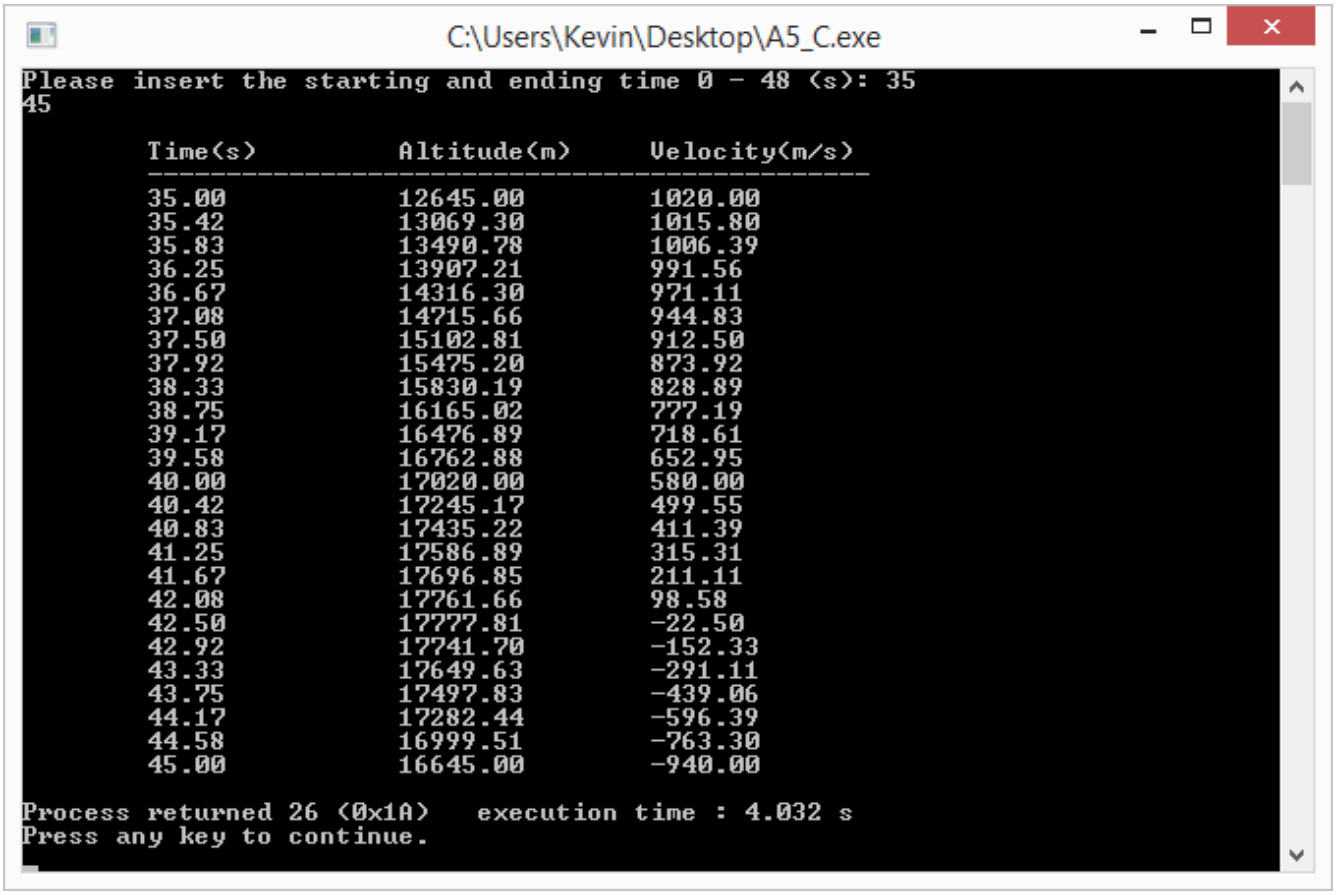
1) The source code of your program (DO NOT insert the source code into your PDF assignment file).
2) Insert the output for the test cases shown in the table on the next page into your assignment file. Also insert an output that shows how invalid input data is treated.

The following table gives the output for three tests cases (i.e. three ranges of time). The tables were computed using and Excel program (see the file *GNG1106A5TestCases.xlsx*).

Entre 2.0 et 5.0			**Entre 1.0 et 44.0**			**Entre 35.0 et 45.0**		
t	**alt(t)**	**v(t)**	**t**	**alt(t)**	**v(t)**	**t**	**alt(t)**	**v(t)**
2.00	6994.08	2720.16	1.00	3951.88	3375.52	35.00	12645.00	1020.00
2.13	7329.26	2642.96	2.79	8958.13	2248.45	35.42	13069.30	1015.80
2.25	7654.86	2566.78	4.58	12131.45	1326.70	35.83	13490.78	1006.39
2.38	7971.00	2491.63	6.38	13824.87	593.70	36.25	13907.21	991.56
2.50	8277.81	2417.50	8.17	14361.72	32.89	36.67	14316.30	971.11
2.63	8575.42	2344.38	9.96	14035.67	-372.30	37.08	14715.66	944.83
2.75	8863.95	2272.27	11.75	13110.71	-638.42	37.50	15102.81	912.50
2.88	9143.53	2201.16	13.54	11821.15	-782.05	37.92	15475.20	873.92
3.00	9414.28	2131.04	15.33	10371.62	-819.75	38.33	15830.19	828.89
3.13	9676.33	2061.91	17.13	8937.08	-768.08	38.75	16165.02	777.19
3.25	9929.80	1993.77	18.92	7662.81	-643.61	39.17	16476.89	718.61
3.38	10174.81	1926.61	20.71	6664.41	-462.89	39.58	16762.88	652.95
3.50	10411.49	1860.42	22.50	6027.81	-242.50	40.00	17020.00	580.00
3.63	10639.96	1795.20	24.29	5809.26	1.00	40.42	17245.17	499.55
3.75	10860.33	1730.94	26.08	6035.32	251.06	40.83	17435.22	411.39
3.88	11072.73	1667.63	27.88	6702.90	491.09	41.25	17586.89	315.31
4.00	11277.28	1605.28	29.67	7779.20	704.55	41.67	17696.85	211.11
4.13	11474.09	1543.87	31.46	9201.77	874.87	42.08	17761.66	98.58
4.25	11663.29	1483.40	33.25	10878.47	985.47	42.50	17777.81	-22.50
4.38	11844.98	1423.87	35.04	12687.50	1019.81	42.92	17741.70	-152.33
4.50	12019.29	1365.26	36.83	14477.34	961.31	43.33	17649.63	-291.11
4.63	12186.34	1307.58	38.63	16066.85	793.41	43.75	17497.83	-439.06
4.75	12346.22	1250.81	40.42	17245.17	499.55	44.17	17282.44	-596.39
4.88	12499.07	1194.95	42.21	17771.78	63.16	44.58	16999.51	-763.30
5.00	12645.00	1140.00	44.00	17376.48	-532.32	45.00	16645.00	-940.00


C:\Users\Kevin\Desktop\A5_C.exe
Please insert the starting and ending time 0 - 48 (s): -1
-10
The start time must be between 0 and 48 seconds
Please insert a valid start time: 50
The start time must be between 0 and 48 seconds
Please insert a valid start time: 2
The end time must be between 0 and 48 seconds
Please insert a valid end time: 1000
The end time must be between 0 and 48 seconds
Please insert a valid end time: 5
Time(s) Altitude(m) Velocity(m/s)
2.00 6994.08 2720.16
2.13 7329.26 2642.96
2.25 7654.86 2566.78
2.38 7971.00 2491.63
2.50 8277.81 2417.50
2.63 8575.42 2344.38
2.75 8863.95 2272.27
2.88 9143.53 2201.16
3.00 9414.28 2131.04
3.13 9676.33 2061.91
3.25 9929.80 1993.77
3.38 10174.81 1926.61
3.50 10411.49 1860.42
3.63 10639.96 1795.20
3.75 10860.33 1730.94
3.88 11072.73 1667.63
4.00 11277.28 1605.28
4.13 11474.09 1543.87
4.25 11663.29 1483.40
4.38 11844.98 1423.87
4.50 12019.29 1365.26
4.63 12186.34 1307.58
4.75 12346.22 1250.81
4.88 12499.07 1194.95
5.00 12645.00 1140.00
Process returned 25 (0x19) execution time : 28.072 s
Press any key to continue.


C:\Users\Kevin\Desktop\A5_C.exe
Please insert the starting and ending time 0 - 48 (s): 1
44
Time(s) Altitude(m) Velocity(m/s)
1.00 3951.88 3375.52
2.79 8958.13 2248.45
4.58 12131.45 1326.70
6.38 13824.87 593.70
8.17 14361.72 32.89
9.96 14035.67 -372.30
11.75 13110.71 -638.42
13.54 11821.15 -782.05
15.33 10371.62 -819.75
17.13 8937.08 -768.08
18.92 7662.81 -643.61
20.71 6664.41 -462.89
22.50 6027.81 -242.50
24.29 5809.26 1.00
26.08 6035.32 251.06
27.88 6702.90 491.09
29.67 7779.20 704.55
31.46 9201.77 874.87
33.25 10878.47 985.47
35.04 12687.50 1019.81
36.83 14477.34 961.31
38.63 16066.85 793.41
40.42 17245.17 499.55
42.21 17771.78 63.16
44.00 17376.48 -532.32
Process returned 26 (0x1A) execution time : 6.326 s
Press any key to continue.


C:\Users\Kevin\Desktop\A5_C.exe
Please insert the starting and ending time 0 - 48 (s): 35
45
Time(s) Altitude(m) Velocity(m/s)
35.00 12645.00 1020.00
35.42 13069.30 1015.80
35.83 13490.78 1006.39
36.25 13907.21 991.56
36.67 14316.30 971.11
37.08 14715.66 944.83
37.50 15102.81 912.50
37.92 15475.20 873.92
38.33 15830.19 828.89
38.75 16165.02 777.19
39.17 16476.89 718.61
39.58 16762.88 652.95
40.00 17020.00 580.00
40.42 17245.17 499.55
40.83 17435.22 411.39
41.25 17586.89 315.31
41.67 17696.85 211.11
42.08 17761.66 98.58
42.50 17777.81 -22.50
42.92 17741.70 -152.33
43.33 17649.63 -291.11
43.75 17497.83 -439.06
44.17 17282.44 -596.39
44.58 16999.51 -763.30
45.00 16645.00 -940.00
Process returned 26 (0x1A) execution time : 4.032 s
Press any key to continue.