

GNG1106 Winter 2019 - Assignment 4

Available: Feb. 11

Due: Feb 17, 23:59

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 file in a PDF file (this allows you to use your favorite editor to create the PDF file). For question 1, insert the programming models for parts (a) and (b) 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 Question 2, insert in your assignment file the source code (take care of its appearance), and capture the output from running the program for all test cases. Also submit your **source code** file (.c file) for question 2.
- Place all your files (PDF file and C source code files) in a directory A4_xxxxxxx where xxxxxxx and your student number.
- Zip your PDF document and the C source files in a zip file with the name A4_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 4” to reach the assignment 4 submission folder. You can also select the Assignment tab to see the Assignment folder pages. The Brightspace video “Assignments” (found in the page https://documentation.brightspace.com/EN/le/assignments/learner/submit_assignments.htm) 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 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 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: 15 marks
- Question 2: 15 marks

Question 1 (15 marks)

(a) (10 marks) The following programming model contains in its code memory the indicated C program composed of 2 functions. You will be showing how the working memory is used during the execution of the two functions. Each piece of working memory is associated to a function using a pair of lines. (Note: the first pair of lines associates the piece of working memory allocated to the function **main** and the second pair of lines associates the piece allocated to the function **compute_weight**). Show how the given C program affects the contents of the working memory:

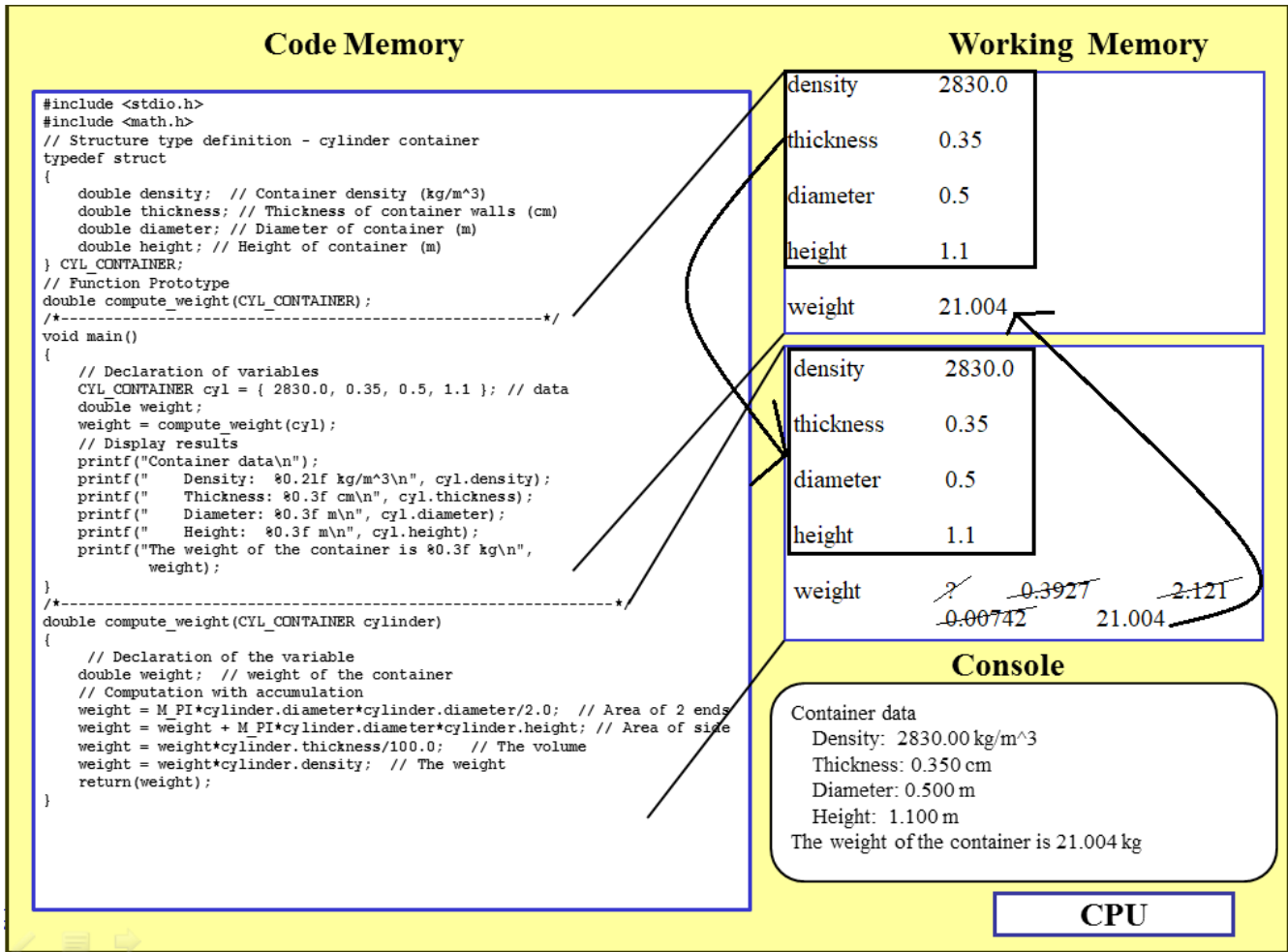
- Show how the structure is organized in the working memory for each function.
- Show the values are assigned to the members of the structure. Be sure to show all values that are assigned and replaced. Record successive assignments to variables/parameters as follows:

Variable name ? \ 2, 6, 4, 10

- Show the values are assigned to the members of the structure. Be sure to show all values that are assigned and replaced. Record successive assignments to structure members as follows:

..

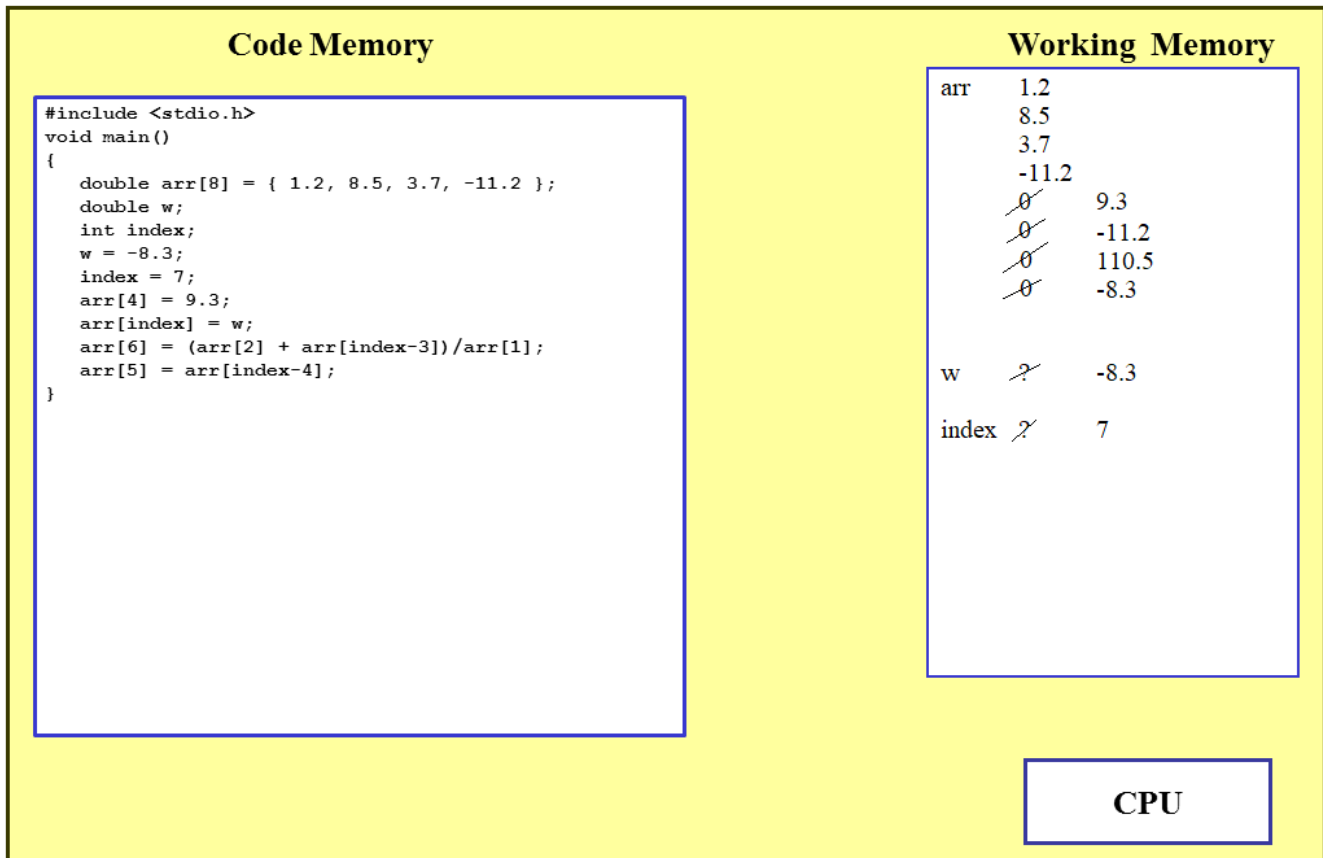
- Using arrows show how values are copied between the working memory allocated to the function **main** and the working memory allocated to the function **compute_weight**. For copying the contents of a structure variable, you need only used a single arrow.
- In the console window show the output of the program.



Note: You do NOT need to show how the CPU interacts with memory.

- (b) (5 marks) The following programming model contains in its code memory the indicated C program composed of a single function, main. You will be showing how the working memory is used during the execution of main. Show how the given C program affects the contents of the working memory:
- Show the values are assigned to the variables and array elements. Be sure to show all values that are assigned and replaced. Record successive assignments to variables/parameters as follows:

Variable name? \ 2, 6, 4, 10



Question 2 (15 marks)

The average velocity of water in a rectangular open channel can be calculated using the following Manning's equation.

$$U = \frac{\sqrt{S}}{n} \left(\frac{B}{B + 2H} \right)^{2/3}$$

where U is the average velocity of the water (m/s),
S is the channel slope
n is the roughness coefficient (s/m^{1/3})
B is the width (m)
H is the depth of the water (m)

Develop a program that requests from the user characteristics of the channel and display in table form how the average velocity of the water varies with its depth in the channel. Use the following guidelines in developing your program.

- Use a C structure type:
 - Define a structure type `CHANNEL` that contains with members for the values for the open channel, that is, members `name` that holds a string for the name of the channel, `n` (roughness coefficient), `slope` (value of S), `width` (value of B), and `maxDepth` that holds the maximum water depth possible in the channel.
- In the main function:
 - Obtain from the user the values to fill all members of a structure variable of type `CHANNEL`. Use the function `fgets` to initialise the member `name` (this function will place the complete line typed by the user including spaces as opposed to the `scanf` function that stops when it encounters a space). For the value of the other members of the structure variable (that are of type `double`), used the function `getPositiveValue` from the *CylinderVolumeLab4* project to ensure that a positive value is stored in each of these members.
 - Call the `displayTable` function to display a table that shows how the average speed of the water changes with its depth in the channel. The function will also display the characteristics of the channel.
- Function `displayTable`
 - This function will have one parameter of type `CHANNEL`.
 - Use a symbolic constant to determine the number of lines to display in the table. Use 25 as the value for this constant. Use other symbolic comments where appropriate to avoid magic numbers.
 - In a first step, the function shall fill one or two arrays with the values of the depth and average velocity to display; you have the option of using one 2D array or two 1D arrays. The function obtains values for the average speed by calling the function `computeVelocity`. Note that Manning's equation cannot be applied to a depth with value 0. Thus the displayed table does not start at a depth of 0, but at the increment value used to increment the depth of the water.

- In a second step, the function displays on the console first the characteristics of the channel followed by a table of 25 lines that shows how the average speed of the water changes with its depth. The following shows an example of the desired output. Be sure to properly format the values.

```

D:\UofO\Courses\CurrentCourses\GNG1106\Fall2018\Assi...
Give the name of the channel: Channel 1
Give the coefficient of roughness: 0.035
Give the slope: 0.0001
Give the channel width: 10
Give the maximum depth of the channel: 4.2

Channel data for "Channel 1"
Coefficient of roughness: 0.0350
Slope: 0.00010
Width: 10.00
Maximum depth: 4.20
-----
Depth          Average velocity
-----
0.17          0.9180
0.34          0.5661
0.50          0.4232
0.67          0.3424
0.84          0.2894
1.01          0.2515
1.18          0.2228
1.34          0.2002
1.51          0.1819
1.68          0.1667
1.85          0.1538
2.02          0.1428
2.18          0.1333
2.35          0.1249
2.52          0.1175
2.69          0.1109
2.86          0.1050
3.02          0.0997
3.19          0.0948
3.36          0.0904
3.53          0.0864
3.70          0.0826
3.86          0.0792
4.03          0.0760
4.20          0.0731

Process returned 0 (0x0)   execution time :
Press any key to continue.

```

- Function `computeVelocity`
 - This function has two parameters, one of type `double` which gives the depth of the water, and the second of type `CHANNEL` which contains the characteristics of the channel.
 - It computes the average velocity of the water using Manning's equation and returns this value.
- The following table gives three test cases to be used for testing your program.

Name	Channel 1	Name	Channel 2	Name	Channel 3
------	-----------	------	-----------	------	-----------

Roughness n (s/m^{1/3})	0.035
Slope (m)	0.0001
Width(m)	10
Max. Depth (m)	4.2
	Average Velocity (m/s)
Depth (m)	
0.1680	0.917961
0.3360	0.566077
0.5040	0.423161
0.6720	0.342380
0.8400	0.289368
1.0080	0.251450
1.1760	0.222759
1.3440	0.200172
1.5120	0.181859
1.6800	0.166669
1.8480	0.153840
2.0160	0.142843
2.1840	0.133301
2.3520	0.124935
2.5200	0.117535
2.6880	0.110939
2.8560	0.105020
3.0240	0.099677
3.1920	0.094829
3.3600	0.090410
3.5280	0.086363
3.6960	0.082644
3.8640	0.079214
4.0320	0.076040
4.2000	0.073095

Roughness n (s/m^{1/3})	0.0013
Slope (m)	0.0032
Width (m)	2
Max. Depth (m)	11.5
	Average Velocity (m/s)
Depth (m)	
0.4600	56.740241
0.9200	29.778691
1.3800	19.693713
1.8400	14.450211
2.3000	11.266883
2.7600	9.146052
3.2200	7.641631
3.6800	6.524830
4.1400	5.666601
4.6000	4.988857
5.0600	4.441703
5.5200	3.991847
5.9800	3.616268
6.4400	3.298570
6.9000	3.026780
7.3600	2.791958
7.8200	2.587308
8.2800	2.407573
8.7400	2.248630
9.2000	2.107203
9.6600	1.980655
10.1200	1.866844
10.5800	1.764015
11.0400	1.670711
11.5000	1.585720

Roughness n (s/m^{1/3})	0.17
Slope (m)	0.041
Width (m)	40
Max. Depth (m)	1.5
	Average Velocity (m/s)
Depth (m)	
0.0600	7.756067
0.1200	4.876297
0.1800	3.713932
0.2400	3.059721
0.3000	2.631589
0.3600	2.325820
0.4200	2.094561
0.4800	1.912415
0.5400	1.764548
0.6000	1.641663
0.6600	1.537612
0.7200	1.448154
0.7800	1.370260
0.8400	1.301702
0.9000	1.240806
0.9600	1.186282
1.0200	1.137124
1.0800	1.092530
1.1400	1.051856
1.2000	1.014577
1.2600	0.980258
1.3200	0.948540
1.3800	0.919119
1.4400	0.891740
1.5000	0.866183

The answer to this question should provide in the assignment file file:

- 1) The source code to your program (also insert the source code into your assignment file).
- 2) The output for each test case in the above table into your assignment file.

```
main.c [Assignment 4 q2] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c x
1 #include <stdio.h>
2 #include <math.h>
3 #define NUM_LINES 25
4 #define EX 0.66667
5 typedef struct
6 {
7     double n;
8     double slope;
9     double width;
10    double maxDepth;
11
12    char name[NUM_LINES];
13 } CHANNEL;
14
15 double getPosValue();
16 double calculateVelocity(CHANNEL, double);
17
18 int main(void)
19 {
20     CHANNEL channel;
21     double T;
22     printf("Give the name of the channel: ");
23     fgets(channel.name, NUM_LINES, stdin);
24     printf("Give the coefficient of roughness: ");
25     channel.n=getPosValue();
26     printf("Give the slope: ");
27     channel.slope=getPosValue();
28     printf("Give the channel width: ");
29     channel.width=getPosValue();
30     printf("Give the maximum depth of the channel: ");
31     channel.maxDepth=getPosValue();
32     printf("\nChannel data for: %s", channel.name);
33     printf(" Coefficient of roughness: %.4lf", channel.n);
34     printf("\n Slope: %.5lf", channel.slope);
35     printf("\n Width: %.2lf", channel.width);
36 }
```

C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 54, Col 32, Pos 1553 Insert Read/Write default 7:50 PM 14/2/19

```
main.c [Assignment 4 q2] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c x
34 printf(" Coefficient of roughness: %.4f", channel.n);
35 printf("\n Slope: %.5f", channel.slope);
36 printf("\n Width: %.2f", channel.width);
37 printf("\n Maximum depth: %.2f\n", channel.maxDepth);
38
39 double depths[NUM_LINES];
40 double velocities[NUM_LINES];
41 double depth;
42 double inc;
43 int ix;
44 printf("%10s %11s\n-----\n", "Depth", "Average velocity");
45 inc = channel.maxDepth/(NUM_LINES);
46 depth=inc;
47 for(ix = 0; ix < NUM_LINES; ix++)
48 {
49     depths[ix] = depth;
50     velocities[ix] = calculateVelocity(channel, depth);
51     depth += inc;
52 }
53 for(ix = 0; ix < NUM_LINES; ix++)
54     printf("%10.2f %10.4f\n", depths[ix], velocities[ix]);
55 return(0);
56
57 }
58 double getPosValue()
59 {
60     double v;
61     do
62     {
63         scanf("%lf", &v);
64         if(v <= 0.0)
65             printf("The value must be greater than zero: \n");
66     }
67     while(v <= 0.0);
68     return(v);
69 }
```

C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 54, Col 32, Pos 1553 Insert Read/Write default 7:50 PM 14/2/19

```
main.c [Assignment 4 q2] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
main.c x
46     depth=inc;
47     for(ix = 0; ix < NUM_LINES; ix++)
48     {
49         depths[ix] = depth;
50         velocities[ix] = calculateVelocity(channel, depth);
51         depth += inc;
52     }
53     for(ix = 0; ix < NUM_LINES; ix++)
54         printf("%10.2lf %10.4f\n", depths[ix], velocities[ix]);
55     return(0);
56 }
57
58 double getPosValue()
59 {
60     double v;
61     do
62     {
63         scanf("%lf",&v);
64         if(v <= 0.0)
65             printf("The value must be greater than zero: \n");
66     }
67     while(v <= 0.0);
68     return(v);
69 }
70 double calculateVelocity(CHANNEL channel, double depth)
71 {
72     double V, num, den;
73     num=channel.width/depth;
74     den=channel.width+2*depth;
75     V=num/den;
76     V=pow(V, EX);
77     V=V*(sqrt(channel.slope)/channel.n);
78     return(V);
79 }
80 }
```

C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 54, Col 32, Pos 1553 Insert Read/Write default 7:50 PM 14/2/19

```
"C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\bin\Debug\Assignment 4 q2.exe"
Give the name of the channel: Channel 1
Give the coefficient of roughness: 0.0350
Give the slope: .0001
Give the channel width: 10
Give the maximum depth of the channel: 4.2000

Channel data for: Channel 1
Coefficient of roughness: 0.0350
Slope: 0.00010
Width: 10.00
Maximum depth: 4.20
Depth      Average velocity
-----
0.17      0.9180
0.34      0.5661
0.50      0.4232
0.67      0.3424
0.84      0.2894
1.01      0.2514
1.18      0.2228
1.34      0.2002
1.51      0.1819
1.68      0.1667
1.85      0.1528
2.02      0.1428
2.18      0.1333
2.35      0.1249
2.52      0.1175
2.69      0.1109
2.86      0.1050
3.02      0.0997
3.19      0.0948
3.36      0.0904
3.53      0.0864
3.70      0.0826
3.86      0.0792
4.03      0.0760
4.20      0.0731

Process returned 0 (0x0)   execution time : 18.839 s
Press any key to continue.
```

```
"C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\bin\Debug\Assignment 4 q2.exe"
Give the name of the channel: Channel 2
Give the coefficient of roughness: 0.0013
Give the slope: 0.0032
Give the channel width: 2
Give the maximum depth of the channel: 11.5

Channel data for: Channel 2
Coefficient of roughness: 0.0013
Slope: 0.00320
Width: 2.00
Maximum depth: 11.50
Depth      Average velocity
-----
0.46      56.7403
0.92      29.7786
1.38      19.6936
1.84      14.4501
2.30      11.2668
2.76      9.1460
3.22      7.6416
3.68      6.5248
4.14      5.6665
4.60      4.9888
5.06      4.4417
5.52      3.9918
5.98      3.6162
6.44      3.2985
6.90      3.0267
7.36      2.7919
7.82      2.5873
8.28      2.4075
8.74      2.2486
9.20      2.1072
9.66      1.9806
10.12     1.8668
10.58     1.7640
11.04     1.6707
11.50     1.5857

Process returned 0 (0x0)   execution time : 18.987 s
Press any key to continue.
```

```
Select "C:\UNIVERSITY\GNG 1106\Assignments\Assignment 4\bin\Debug\Assignment 4 q2.exe"
Give the name of the channel: Channel 3
Give the coefficient of roughness: 0.17
Give the slope: 0.041
Give the channel width: 40
Give the maximum depth of the channel: 1.5

Channel data for: Channel 3
Coefficient of roughness: 0.1700
Slope: 0.04100
Width: 40.00
Maximum depth: 1.50
-----
Depth      Average velocity
-----
0.06      7.7561
0.12      4.8763
0.18      3.7140
0.24      3.0597
0.30      2.6316
0.36      2.3258
0.42      2.0946
0.48      1.9124
0.54      1.7646
0.60      1.6417
0.66      1.5376
0.72      1.4492
0.78      1.3783
0.84      1.3017
0.90      1.2408
0.96      1.1863
1.02      1.1371
1.08      1.0925
1.14      1.0519
1.20      1.0146
1.26      0.9803
1.32      0.9485
1.38      0.9191
1.44      0.8917
1.50      0.8662

Process returned 0 (0x0)   execution time : 18.076 s
Press any key to continue.
```