GNG1106 Fall 2019 - Assignment 4

Available: Oct 11
Due: Oct 23, 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 hand draw your programming model figures, take photos of them and insert them into your document. For Question 2, insert in your assignment file the source code (take care in its appearance), and capture the output from running the program for all test cases. Also submit your **source code** files for question 2.
- Place all your files (PDF file and C source code files) in a directory A4_xxxxxxxx where xxxxxxxx et 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.
- Submit the zip file before the assignment deadline via Brightspace. In Brightspace, Click "Assignments" on the top navigation bar to access the assignment submission page. 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.
- Do start the assignment soon and do not wait until the last minute.

Marking Scheme (total 100 marks)

- Question 1: 30 marks
- Question 2: 40 marks
- Question 3: 30 marks

Question 1 (30 marks). Trace the following two programs in the programming models. For required drawing format of the programming model, refer to "programmingModelFormat.pdf", which is given together with the lecture slides.

(a)

```
#include <stdio.h>
#include <math.h>
// Structure type definition - cylinder container
typedef struct
{
    double density;  // Container density (kg/m^3)
    double thickness; // Thickness of container walls (cm)
    double diameter; // Diameter of container (m)
    double height; // Height of container (m)
} CYL_CONTAINER;
// Function Prototype
double compute_weight(CYL_CONTAINER);
/*-----------------------------------------------------*/
void main()
{
    // Declaration of variables
    CYL_CONTAINER cyl = { 2830.0, 0.35, 0.5, 1.1 }; // data
    double weight;
    weight = compute_weight(cyl);
    // Display results
    printf("Container data\n");
    printf("    Density:  %0.2lf kg/m^3\n", cyl.density);
    printf("    Thickness: %0.3f cm\n", cyl.thickness);
    printf("    Diameter: %0.3f m\n", cyl.diameter);
    printf("    Height:  %0.3f m\n", cyl.height);
    printf("The weight of the container is %0.3f kg\n",
            weight);
}
/*-------------------------------------------------------------*/
double compute_weight(CYL_CONTAINER cylinder)
{
     // Declaration of the variable
    double weight;  // weight of the container
```

```
    // Computation with accumulation
    weight = M_PI*cylinder.diameter*cylinder.diameter/2.0;  // Area of 2
ends
    weight = weight + M_PI*cylinder.diameter*cylinder.height; // Area of
side
    weight = weight*cylinder.thickness/100.0;   // The volume
    weight = weight*cylinder.density;  // The weight
    return(weight);
}
```

(b)

```
#include <stdio.h>
void main()
{
   double arr[8] = { 1.2, 8.5, 3.7, -11.2 };
   double w;
   int index;
   w = -8.3;
   index = 7;
   arr[4] = 9.3;
   arr[index] = w;
   arr[6] = (arr[2] + arr[index-3])/arr[1];
   arr[5] = arr[index-4];
}
```

Question 2 (40 marks)

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

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

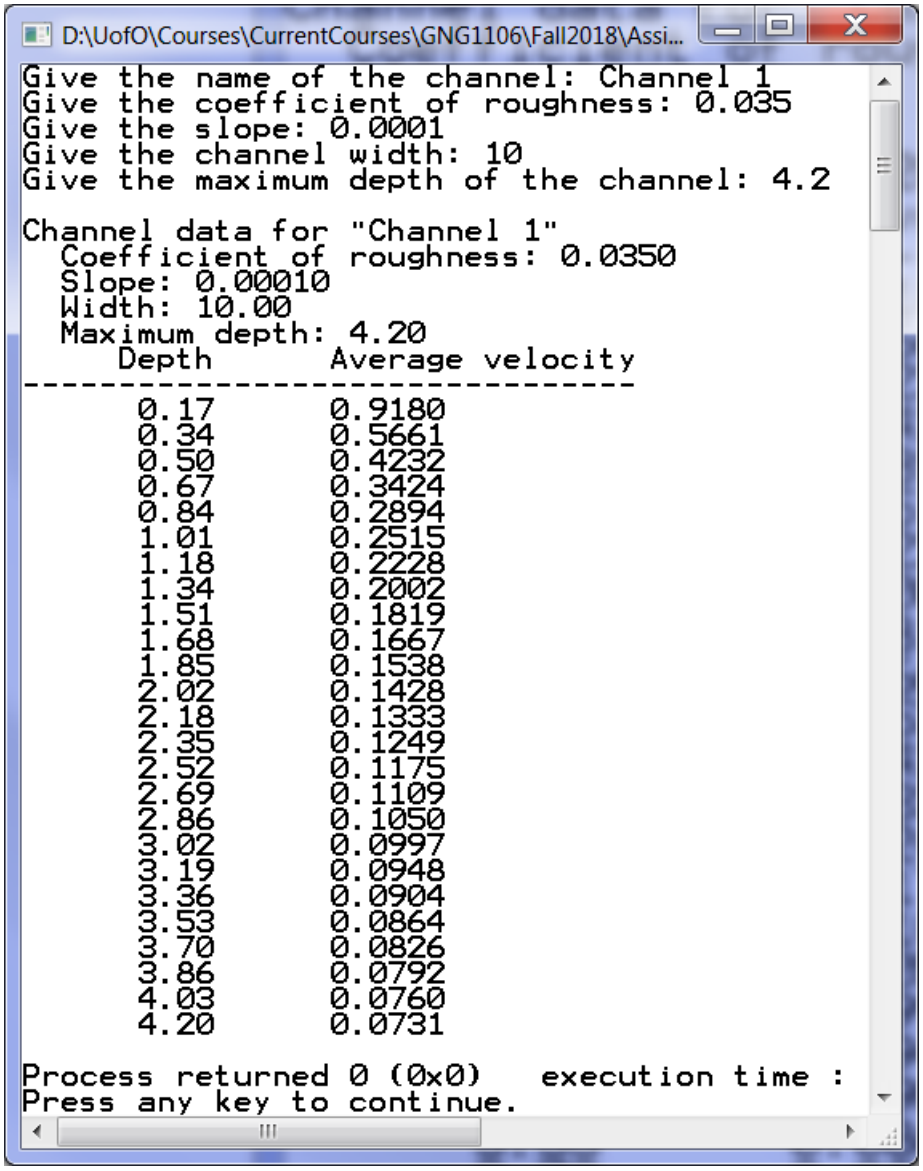
where U is the average velocity of the water (m/s),
S is the channel slope
n is the roughness coefficient ($s/m^{1/3}$)
B is the width (m)
H is the depth of the water (m)

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.



- 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
Roughness n (s/m^1/3)	0.035
Slope (m)	0.0001
Width(m)	10
Max. Depth (m)	4.2
Depth (m)	**Average Velocity (m/s)**
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

Name	Channel 2
Roughness n (s/m^1/3)	0.0013
Slope (m)	0.0032
Width (m)	2
Max. Depth (m)	11.5
Depth (m)	**Average Velocity (m/s)**
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

Name	Channel 3
Roughness n (s/m^1/3)	0.17
Slope (m)	0.041
Width (m)	40
Max. Depth (m)	1.5
Depth (m)	**Average Velocity (m/s)**
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.

Question 3 (30 marks)

The following is a block of C code (intentionally written without indentation). Answer the questions without running the program.

```
if ((x>y) && (y >0))
x=x+y;
y++;
if (x >10)
y=x*y;
else
y=x+x*x+y;
printf (" the values of  x  and  y  are respectively %d  and  %d\n", x, y);
```

What will this block of code print if right before the block int-typed variables x and y take respectively the following pair of values?

1. 5 and 3
2. 7 and 4
3. 5 and -1
4. 11 and 12;