The University of Nottingham

DEPARTMENT OF MECHANICAL, MATERIALS AND MANUFACTURING ENGINEERING

A LEVEL 3 MODULE, AUTUMN SEMESTER 2022-2023

COMPUTER ENGINEERING AND MECHATRONICS

Time allowed 2 hours

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer ALL questions

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

ADDITIONAL MATERIAL: Letters.txt

INFORMATION FOR INVIGILATORS:

Question papers should be collected in at the end of the exam – do not allow candidates to take copies from the exam room.

SECTION A

Answer this section by writing programs in the C language using the GCC compiler within VSCode

Note: All questions in this section that require numerical values to be input by the user should be designed and coded to reject invalid values (with a suitable error message displayed in such cases). No global variables should be used in any programs.

For all questions of **Section A**:

- Code should compile without warnings and use appropriate comments and variable definitions and names. Marks allocated for this cover all questions. [6]

1. The square root of a number can be calculated numerically by an iterative method using the equation below.

$$x_{n+1} = 0.5(x_n + \frac{A}{x_n})$$

Where x_n and x_{n+1} are consecutive estimates of the square root of a number, A.

You are to develop a program that:

- Calculates the square root of a positive number which is input by the user.
- Stops the calculation when the solution has converged to within 0.0001%.
- Displays a message showing the original number and the square root, both to 5 decimal places of accuracy.
- Displays the number of iterations to arrive at the solution.

[12]

2. (a) Develop a function **external to main()** that calculates area of a regular sided polygon using the equation below.

$$Area = \frac{L^2 n}{4\tan(\pi/n)}$$

Where L = length of the side in m
n = number of sides

The function should meet the following design criteria:

- The function is to return a value of 1 if area is calculated successfully.
- The function is to return a value of 0 if the number of sides is less than 3 or the length is less than 0.
- The length can be a non-integer value.
- Area is to be returned via the argument list (using a suitable variable).

[8]

Continued on next page

(b) Write a program (i.e. the main() function) to meet the following design criteria:

- The user should be prompted for input of length and number of sides in the polygon.
- Use the function developed in (a) to calculate the area of the polygon.
- The number of sides, length and area are to be displayed with an appropriate message to 2 decimal places of accuracy (where appropriate), including units.
- An error message should be displayed if the function returned a value of 0 (i.e. the inputs were invalid).

The following data can be used to test your program:

Sides	Length (m)	Area (m^2)
4	10	100
6	-22.2	184.82
-1	2	Error
3	-1	Error

[8]

3. The file 'Letters.txt' provided contains some text. Write a program which will read the text from the file one character at a time and then display information about the characters in one of three formats as given below:

Format 1: Display a string which shows which character was read.
e.g. "The character read was t".

Format 2: Display a string which shows the ASCII value of the character.
e.g. "The ASCII value of the character is 27".

Format 3: Display a string which gives both the character and its ASCII value. e.g. "The ASCII value of d is 100".

The user should be prompted to input the name of the input file and the format required (the same format is to be used for all characters for a single run of the program).

[16]

SECTION B

4. (a) Explain how you would take measurements of the following quantities into a microcontroller such as an Arduino Mega. In each case, identify the following: **1**) the kind of measuring device and any ancillary equipment needed, 2) the form of the electrical signal, 3) the kind of interface(s) needed to get the data into the microcontroller and 4) any additional features required to maintain the accuracy of the system.

   i) The position of the saddle of a lathe along the lathe bed, for example as measured within a digital readout system. [5]

   ii) A temperature in the range 0°C to 1200°C. [5]

   iii) The applied tensile force on a metallic rod in a machine. [5]

   (b) Explain the concept of an open collector logic gate, comparing and contrasting it with a more conventional (totem-pole) arrangement. Within your answer explain the circumstances in which an open collector circuit is used in preference to a totem pole circuit. You are encouraged to illustrate your answer with simplified circuit diagrams which illustrate the principle but are not expected to draw detailed diagrams of actual circuits. [5]

   (c) Often it is needed to get data into the PC's data bus and into program from external device (e.g., encoder). The external device will put either 5V or 0V on (typically) eight wires (parallel data). This data will need to be transferred to the CPU through data bus. When the CPU uses the data bus for other purposes (e.g., sending signal to output ports), external signals would swamp/corrupt data. Explain how this problem is solved in microprocessor and support your answer with simplified circuit diagrams. [6]

5. (a) Explain how one of the timer-counter units in an AVR microcontroller (such as that used in an Arduino Uno or Mega 2560) is used to generate a PWM signal. Illustrate your answer with a diagram representing how one of the values stored within the timer-counter varies with time and relate this to the way in which the PWM signal is generated. You are not expected to remember the names of particular registers, but you are expected to explain the concept of how the PWM signal is generated. [11]

   (b) If it is required to replace a DC motor (embedded with an optical encoder) with a bi-polar stepper motor, use a schematic diagram to show how a DC motor driver (e.g., a dual H-bridge) can be used to drive the stepper motor using an Arduino Mega. Support your answer with diagram(s). [6]

   (c) Briefly explain the concept of a flash analoge to digital converter. Support your answer with diagram(s). [7]