

The University of Nottingham Ningbo China

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A LEVEL 1 MODULE, 2022-2023

EEEE1044 Introduction to Software Engineering and Programming

Time allowed: **TWO Hours**

Candidates may log in to computers and test CodeBlocks and sign their desk card but must NOT write anything else until the start of the examination period is announced.

Answer ALL Questions

Only a calculator from approved list A (or one functionally equivalent) may be used in this examination.

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

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

Do read the Exam Procedure available on desktop before starting the exam.

ADDITIONAL MATERIAL: Exam Procedure

INFORMATION FOR INVIGILATOR:

None.

Q1. Develop a programme that meets all of the following requirements.

- a) Generate **ONE** random integer number in the range from 0 to 24 (inclusive).
- b) Display the appropriate message for the random number as specified in Table Q1.

Table Q1

| | |
|----------------------------------|---------------|
| From 0 to 7 and from 22 to 24 | Bed Time |
| 8, 9, 10, 11, 12 | Morning! |
| 13, 14 | Get Fuel! |
| 15, 16, 17, 18 | Keep fighting |
| 19, 20, 21 | Fun Time |

[6 marks]

- Q2. Develop a programme to print a triangular pattern that meets all of the following requirements.
- a) Prompt the user to enter a **positive even** integer, e.g., 2, 4, 10. Check whether the entered input integer satisfies the requirements. If not, prompt the user to input again until a valid input is made.
 - b) Use the input integer as the **row number** of a **triangular pattern**, **odd** lines printed with the asterisk symbol (i.e., '*') and with **even** rows printed with the circumflex symbol (i.e., '^'). Take the integer **6** as an example where the pattern that needs to be printed is shown in Figure Q2.
 - c) Comment your codes properly and use the appropriate indentation.

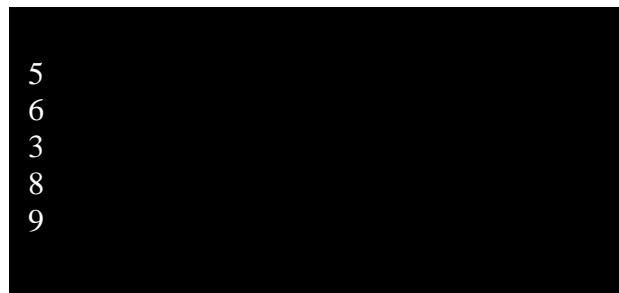
```
*  
^^  
***  
^^^^  
*****  
^^^^^^
```

Figure Q2

[20 marks]

Q3. Develop a programme that meets all of the following requirements.

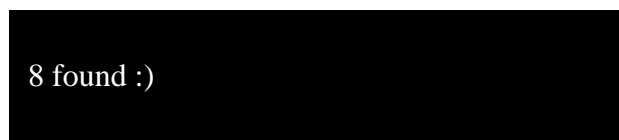
- a) Prompt the user to enter a **5-digit integer**, e.g., 56389, 12345. If the input does not meet the requirements, display an appropriate **error message** and prompt the user to repeat the input **until a valid input** is made.
- b) Pass the integer to an **external function**. In the external function, display the **5 digits separately with one digit per row**. Using the example of entering 56389, the display of the 5 digits is shown in Figure Q3b.



```
5
6
3
8
9
```

Figure Q3b

- c) If **one or more 8s** are present in the 5 digits, display the message **8 found :)**, as shown in Figure Q3c. Perform this task using **the same external function** as in step b). *Note: The message **should be displayed only once, regardless of how many 8s are included in the 5-digit integer**.*



```
8 found :)
```

Figure Q3c

- d) Comment your codes properly and use the appropriate indentation.

[24 marks]

Q4. Develop a programme for collecting the marks of a group of students that meets all of the following requirements.

- a) Declare **an integer array of size 10** to store the marks.
- b) Use a **loop** to prompt the user to input 10 marks. For each mark entered by the user, the programme should **check** if the mark is in the range **from 0 to 100 (both inclusive)**. If the entered mark falls outside the range, an appropriate **error message** should be displayed, and the invalid mark **should not** be stored in the integer array.
- c) After 10 valid marks have been entered, the program should display the **message** shown in Figure Q2c

```
*****
Entered marks are:
```

Figure Q2c

- d) After completing step c), the 10 valid marks are displayed with **a field width of 5, left-justified**, and **5 marks per row**. A display example is shown in Figure Q2d. *Note: The marks displayed by your programme do not have to match the integers shown in Figure Q2d.*

```
*****
Entered marks are:
50  65  85  78  100
20  100 45  63  66
```

Figure Q2d

- e) Then display on screen the **average** of the 10 marks, with **a precision of 2 decimal places**.
- f) Comment your codes properly and use the appropriate indentation.

[25 marks]

- Q5. Develop a programme to calculate the total number of credits a student is taking. The text file named “*Credits.txt*” located on your computer desktop contains the credits of the compulsory modules. Develop your programme to meet all of the following requirements.
- a) A student must select **5 optional modules**. Prompt the student to input the credits for each optional module he/she takes using a **loop**. Credits should be within the range **from 5 to 30 (both inclusive)** and should be **divisible by 5**. Verify that the credits input are valid. Save the valid input credits in the *Credits.txt* file **without erasing** the credits of the saved in the document. *Note: Invalid credits should not be saved in the Credits.txt file.*
 - b) Read all credits stored in *Credits.txt* file, calculate the total credits (sum) and display **all credits** and **the total credits** to the screen. *Note: Design the display to be user friendly.*
 - c) Comment your codes properly and use appropriate indentation.

[25 marks]

- END -