The University of Nottingham Ningbo China

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A LEVEL 1 MODULE, 2023-2024

EEEE1044 Introduction to Software Engineering and Programming

Time allowed: **TWO Hours**

Candidates may log in to computers, 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:
IT support before, during and after exam is requested.

Q1. Develop a program which meets all the following requirements.

a) In **function main()**, prompt the user to enter **two integer values**. Using these values as the limits of a ***for*** loop, count between these limits **with an increment of 1**. *Note: no need to check if the inputs are integers or not.*

[**5 marks**]

b) **Find** and **display** the factors for each integer within the limits, using **a function external to main()**. For example, the factors for 6 include 1, 2, 3, and the factors for 10 include 1, 2, 5. Display the numbers and their factors **left-justified** in a table format with a heading of "**Number Factors of the number**".

[**6 marks**]

c) Comment your program properly and use proper indentation.

[**4 marks**]

Q2. Develop a program which calculates the average mark and the fail percentage. The program should meet all the following requirements.

a) Count the number of marks entered by the user until the user **enters -1 to stop**. The **valid** input of integers should be **from 0 to 100** both of which are included. **Invalid inputs shall not be counted**.

[**5 marks**]

b) Keeps a separate count of the failed marks (less than 40).

[**2 marks**]

c) After finish entering marks, the program displays the total count of marks, the average mark on the screen together with the percentage of the failed marks. Display all the results on the screen to **two decimal places of accuracy**. Make the displayed information to be user-friendly.

[**4 marks**]

d) Comment your program properly and use proper indentation.

[**4 marks**]

Q3. Develop a program which meets all the following requirements.

a) In function **main()**, prompt the user to enter an integer **between 1 and 9** (both included). If the user enters any integer out of this range, prompt the user to enter again, **until** a valid number is entered.

[**3 marks**]

b) Use **a function external to main()** to display the multiplication table on the screen using the user-entered integer as row number. The output of each number takes a **width of 6** characters, and should be **right-justified**. Figure Q3 is an example of the output when the user enters **5**.

[**8 marks**]

c) Comment your program properly and use proper indentation.

[**4 marks**]

```
1
2     4
3     6     9
4     8    12    16
5    10    15    20    25
```

Q4. Develop a program which meets all the following requirements.

a) Declare an integer array of **size 25**. The value of each element is generated using **rand()** and the value is within the range **from 1 to 50** (both included).

[**2 marks**]

b) Prompt the user to enter **one integer** within the range **from 1 to 50** (both included) as the lucky number. Keep prompting if the input is invalid **until** a valid integer is entered.

[**2 marks**]

c) **Display** the integer array to the screen in the following format:

1) 5 elements per row;

2) Each element takes a width of 5 spaces;

3) Display is right-justified.

[**3 marks**]

d) Find whether the lucky number is included in this integer array. Display "***Congratulations!***" if the lucky number is included in this array, or "***You are not so lucky!***" if not.

[**4 marks**]

e) Comment your program properly and use proper indentation.

[**4 marks**]

Q5. Develop a program to collect the name and age of staff working in UNNC. The program shall meet all the following requirements.

a) Prompt the user to enter the name and age of each staff.

1) Staff name is entered in a format of "**FirstnameSurname**", for example, JoeSmith, **without** space in between names. There should be **no more than 30 characters** for a name. Extra characters is ignored and not saved by the program.

2) The age of staff shall be within the range **from 20 to 80** (both included). Any entered age outside of this range shall be ignored and the user shall entered again until a valid age is entered for each staff.

3) Staff name and age are saved to a file named "**AgeStatistics.txt**", with one staff name and age per row, an example is shown in Figure Q5. *Note: the displayed names and ages of your program are not required to be the same as shown in Figure Q5*.

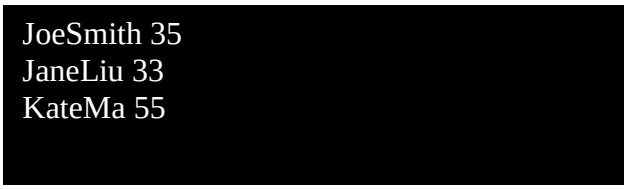

Figure Q5

[**11 marks**]

b) To finish entry, the user press **ctr + z** then press **enter**.

[**2 marks**]

c) Step a) and b) shall be done in function main().

[**2 marks**]

d) Use **a function external to main()** to read all the ages saved in "**AgeStatistics.txt**"; count the staff number; calculate the average age; keep a separate count of staff aged 60 and above (>=60). Return the 3 results (staff number, average age, count ages >= 60) back to **function main()** to display.

[**15 marks**]

e) Make the displayed information to be user-friendly.

[**4 marks**]

f) Comment your program properly and use proper indentation.

[**6 marks**]