

The University of Nottingham Ningbo China

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A LEVEL 1 MODULE, 2021-2022

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 the questions

Only silent, self-contained calculators with a Single-Line Display or Dual-Line Display 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.

At the top of each piece of code you write should be, in addition to any other, a comment statement indicating the question being undertaken and your candidate ID.

You are advised to save your work often, as the machines are purged of files each time they are rebooted.

Save all the programs on desktop first; before the end of exam copy all programs to "Your Name (folder on desktop with your own name)" -> "My documents". Make sure the programs are the up-to-date ones after copying to "My documents".

Q1 Develop a pattern-display program which meets the following requirements:

- Prompt the user to enter **one** integer within the range of **2 ~ 10 (both inclusive)** (*no need to check if the input is integer or not*). Keep prompting if the input is out of this range.
- Display a pattern using the * symbol with a loop function, e.g. *while*, *for*. An example is given below when the input is **4**.

```
*
**
***
****
***
**
*
```

[20 marks]

Marking Scheme

| Item | Mark |
|----------------------------|------|
| Comment | 3 |
| Indentation | 2 |
| Loop checking for input | 5 |
| Display pattern using loop | 10 |

```
//EEEE1044 final exam 21-22
// Q1 pattern display

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int patternsize;
    int line, column;

    puts("Please enter one integer within 2 ~ 10 (both inclusive):");
    scanf("%d", &patternsize );

    // check if input is within the range of 2 ~ 10
    while( ( patternsize < 2 ) || ( patternsize > 10 ) )
    {
        puts("Input is out of range! Enter again (2 ~ 10):");
        scanf("%d", &patternsize );
    } // end while

    // display pattern - top half and longest line
    for( line = 1; line <= patternsize; ++ line )
    {
        // display space
        for( column = 1; column <= ( patternsize - line ); ++ column )
        {
            printf(" ");
        } // end inner for loop
```

```

    // display *
    for( ; column <= patternsize; ++ column )
    {
        printf("*");
    } // end inner for loop

    //start a new line
    puts("");
} // end outer for loop

// display pattern - bottom half
for( line = 1 ; line < patternsize; ++ line )
{
    // display space
    for( column = 1; column <= line; ++ column )
    {
        printf(" ");
    } // end inner for loop

    // display *
    for( ; column <= patternsize; ++ column )
    {
        printf("*");
    } // end inner for loop

    //start a new line
    puts("");
} // end outer for loop

return 0;
}

```

```

Please enter one integer within 2 ~ 10 (both inclusive):

```

```

1

```

```

Input is out of range! Enter again (2 ~ 10):

```

```

11

```

```

Input is out of range! Enter again (2 ~ 10):

```

```

20

```

```

Input is out of range! Enter again (2 ~ 10):

```

```

4

```

```

 *
**
***
****
***
**
*

```

```

Process returned 0 (0x0)   execution time : 7.567 s

```

```

Press any key to continue.

```

Q2 Develop a program which meets the following requirements:

- Prompt the user to enter **an integer with four digits**, e.g. 1202, 2586, 6374 etc., (*no need to check if the input is integer or not*).
- If the integer has more or less than four digits, display a proper **error message**. Keep prompting until a valid integer is entered by the user.
- Display each digit in a single row on the screen. For example, if the user enters an integer of **3564**, the output on the screen should be:

```
3
5
6
4
```

[15 marks]

Marking Scheme

| Item | Mark |
|------------------------------------|------|
| Comment | 3 |
| Indentation | 2 |
| Loop checking for input | 3 |
| Algorithm of isolating each digit | 5 |
| Display format: one digit per line | 2 |

```
//EEEE1044 final exam 21-22
// Q2 isolating digits

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int target, control;
    double i;

    puts("Please input a four-digit integer:");
    scanf("%d", &target );

    // check if input is a 4-digit number
    while( ( target < 1000 ) || ( target > 9999 ) )
    {
        puts("Input is out of range! Enter again (4-digit integer):");
        scanf("%d", &target );
    } // end while

    // isolate the 4 digits and display
    printf("\n\nThe four digits in the entered integer are:\n");
    for( control = 3; control >= 0; --control)
    {
        i = pow(10, control);
        printf("%d\n", target / (int)i );
        target = target % (int)i;
    } // end for

    return 0;
```

```
} // end main
```

Q3 Develop a program which meets the following requirements:

- d) In function **main()**, prompt the user to enter the mark of a student. Keep prompting until **the integer of -1** is entered by the user.
- e) Use **a function external to main()** to **determine** and **display** the ranking of the student's mark. The mark ranking is shown below.
- f) If the input mark is out of the range between 0 ~ 100, a proper **error message** should be displayed within the external function.
- g) When -1 is entered by the user, the program should be terminated immediately **without** displaying the error message.

| <i>Mark Range</i> | <i>Ranking</i> |
|--------------------------------|----------------|
| <i>70 <= mark <= 100</i> | <i>1st</i> |
| <i>60 <= mark < 70</i> | <i>2:1</i> |
| <i>50 <= mark < 60</i> | <i>2:2</i> |
| <i>40 <= mark < 50</i> | <i>3rd</i> |
| <i>0 <= mark < 40</i> | <i>Fail</i> |

[15 marks]

Marking Scheme

| Item | Mark |
|-------------------------------|------|
| Comment | 3 |
| Indentation | 2 |
| -1 to terminate input | 3 |
| Ranking display in ext. func. | 5 |
| No error message for -1 | 2 |

```
Enter student's mark (0~100): 150
out of the range 0 ~ 100
Enter student's mark (0~100): -10
out of the range 0 ~ 100
Enter student's mark (0~100): 30
Fail
Enter student's mark (0~100): 75
1st
Enter student's mark (0~100): 62
2:1
Enter student's mark (0~100): 44
3rd
Enter student's mark (0~100): -1

Process returned 0 (0x0)   execution time : 17.498 s
Press any key to continue.
```

```
//EEEE1044 final exam 21-22
// Q3 mark ranking display
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    int mark;
```

```
    printf("Enter student's mark (0~100): ");
    scanf("%d", &mark );
```

```
while( mark != -1 )
{
    DisplayRanking(mark); // call external function to check and display ranking
    printf("Enter student's mark (0~100): ");
    scanf("%d", &mark );
} // end while

return 0;
} // end main

// external function for mark ranking
void DisplayRanking( int mk )
{
    if( ( mk > 100 ) || ( mk < 0 ) )
        puts("out of the range 0 ~ 100");
    else if ( mk >= 70 )
        puts("1st");
    else if ( mk >= 60 )
        puts("2:1");
    else if ( mk >= 50 )
        puts("2:2");
    else if ( mk >= 40 )
        puts("3rd");
    else
        puts("Fail");

    return;
} // end external function
```

Q4 Develop a program that meets the following requirement:

- Prompt the user to enter **an integer** within the range of **0 ~ 255** (both 0 and 255 are inclusive). If the integer is out of the range, a proper information should be displayed.
- Display on the screen the binary number of the entered integer. For example, the binary number for 0 is 00000000, and the binary number for 255 is 11111111.

[10 marks]

Marking Scheme

| Item | Mark |
|-------------------|------|
| Comment | 2 |
| Indentation | 2 |
| check input range | 2 |
| Convert to binary | 4 |

```
//EEEE1044 final exam 21-22
//Q4 convert decimal to binary

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int decimal, counter = 0;
    char outcome [ 9 ] = "00000000";

    printf("Enter the decimal number between 0 ~ 255: ");
    scanf("%d", &decimal );

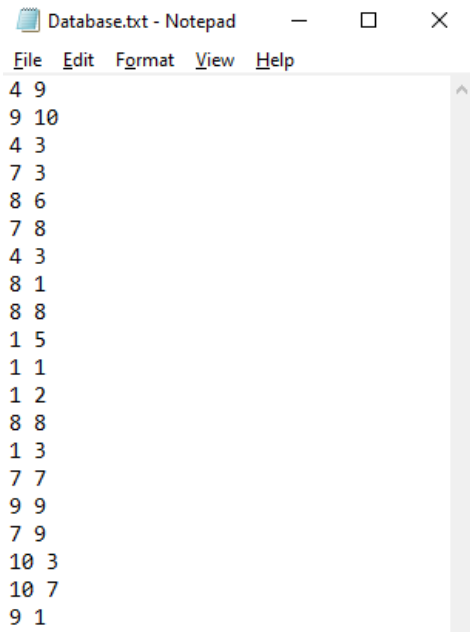
    // check input, to make sure input is within the specified range
    while( ( decimal >255 ) || ( decimal < 0 ) )
    {
        printf("Enter again! Should be within the range of 0 ~ 255: ");
        scanf("%d", &decimal );
    } // end while for input check

    // convert to binary, save results in character array
    while( decimal != 0 )
    {
        outcome[ 7 - counter ] = decimal % 2 + 48;
        decimal = decimal / 2;
        ++counter;
    } // end while for converting

    puts( outcome );
    return 0;
} // end main
```


Q5 Develop a program that meets the following requirement:

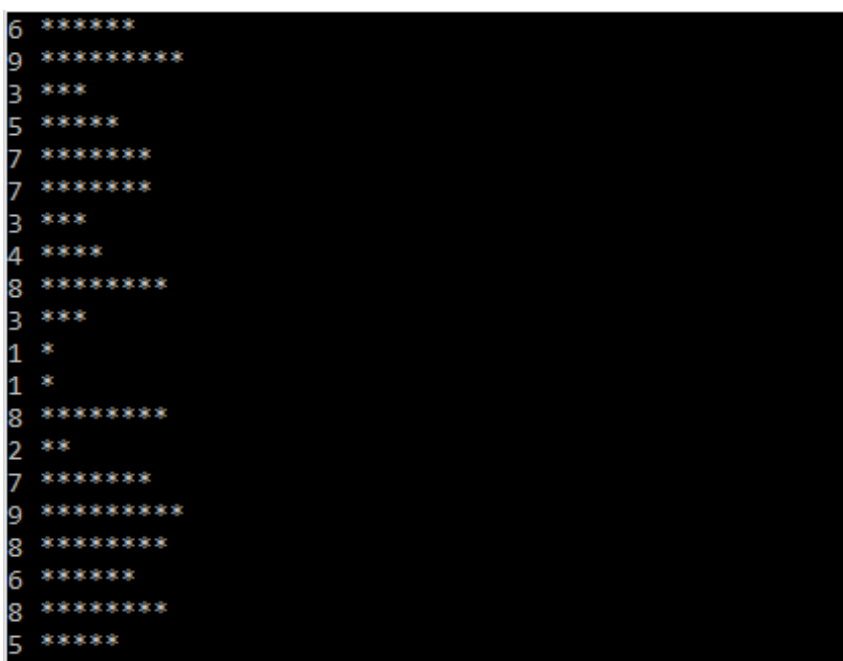
- a) Use the random function to generate **20 pairs** of random integers within the range of **1 ~ 10**. Save these numbers into a file named **“Database.txt”**. When saving the numbers, each pair occupies one line, as the example shows below.



```
Database.txt - Notepad
File Edit Format View Help
4 9
9 10
4 3
7 3
8 6
7 8
4 3
8 1
8 8
1 5
1 1
1 2
8 8
1 3
7 7
9 9
7 9
10 3
10 7
9 1
```

- b) Read from this file **“Database.txt”** in your program. Calculate and display to the screen the average value of each pair of numbers as histogram as shown in Figure Q5b, i.e. calculate the result of $(n1+n2)/2$ using the data type of integer for the result.

*Note: the example below is the output generated based on the random numbers in a), and the random numbers generated by your program should be of **different** combination of numbers from this example).*



```
6 *****
9 *****
3 ***
5 *****
7 *****
7 *****
3 ***
4 *****
8 *****
3 ***
1 *
1 *
8 *****
2 **
7 *****
9 *****
8 *****
6 *****
8 *****
5 *****
```

[10 marks]

Marking Scheme

| Item | Mark |
|-------------------------------|------|
| Comment | 2 |
| Indentation | 2 |
| Writing to file & rand number | 3 |
| Read from file & histogram | 3 |

```
//EEEE1044 final exam 21-22
//Q5 rand number & histogram

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int number1, number2, counter, control;
    FILE *fpnto;

    // open the file for writing random number
    if( ( fpnto = fopen( "Database.txt", "w" ) ) == NULL )
    {
        puts("Fail in opening file!");
    } // end if for check file opening
    else
    {
        srand( time( NULL ) );
        for( counter = 0; counter < 20; ++ counter )
        {
            fprintf( fpnto, "%d %d\n", 1 + rand() % 10, 1 + rand() % 10 );
        } // end for - file writing

        fclose( fpnto );
    } // end else - end writing to file after successful opening

    // open the file to read out the numbers
    if( ( fpnto = fopen( "Database.txt", "r" ) ) == NULL )
    {
        puts("Fail in opening file!");
    } // end if for check file opening
    else
    {
        fscanf( fpnto, "%d %d", &number1, &number2 );

        while( !feof( fpnto ) )
        {
            control = ( number1 + number2 ) / 2;
            printf("%d ", control);
            for( counter = 0; counter < control; ++ counter )
            {
                printf("*");
            } // end printing histogram for one pair
            puts("");

            fscanf( fpnto, "%d %d", &number1, &number2 );
        } // end reading file after complete
    } // end else - end reading from file after successful reading

    return 0;
} // end main
```

Q6 Develop a program to calculate the volume of a cylinder. The radius **R** of the circular face and the height of the cylinder **H** will be entered by the user; the area of the circular face **A** can be calculated using Eq. 1. The volume of the cylinder **V** can be calculated using Eq. 2.

$$A = \pi \cdot R^2 \quad \text{Eq. 1}$$

$$V = H \cdot A \quad \text{Eq. 2}$$

Input **R** and **H** with both value and unit. For example, the input of **R** can be **5 mm** or **0.5 cm**, and the height **H** can be **0.8 m** or **80 cm**. The input units of **m**, **cm** and **mm** should be accepted by the program and all the units should be converted to meters in the calculation; the unit of area **A** and volume **V** should be **m²** and **m³**, respectively.

You are required to develop the program that meets the following requirements:

- Prompt the user to enter the value and the unit for **R**, the radius of the circular face. You should ensure **R > 0**. Otherwise, prompt the user to enter again until a valid **R** is entered. *No need to check if the input unit is one of m, cm and mm, it is assumed that the unit entered by the user meets the requirement of this question.*
- Prompt the user to enter the value and the unit for **H**, the height of the cylinder. You should ensure **H > 0**. Otherwise, prompt the user to enter again until a valid **H** is entered. *No need to check if the input unit is one of m, cm and mm, it is assumed that the unit entered by the user meets the requirement of this question.*
- Use a **function external to main()** to convert the units of **R** and **H** to **meters (m)**.
- Use a **second function external to main()** which accepts the value of **R** in meters and returns the area **A** in **square meters (m²)**.
- Given the values from the user and the value of **A** obtained from (d), use a **third function external to main()** which returns the volume of the cylinder **V** in **cubic meters (m³)**.
- Display the area **A** and the volume **V** to two decimal places of accuracy in function **main()**.

Hint: For a value of R of 120 cm, H of 3000 mm, you should obtain via calculation the following values.

$$\begin{aligned} R &= 120 \text{ cm} = 1.2 \text{ m} \\ H &= 3000 \text{ mm} = 3 \text{ m} \\ A &= \pi r^2 = 3.14 \times (1.2)^2 = 4.52 \text{ m}^2 \\ V &= AH = 4.52 \times 3 = 13.56 \text{ m}^3 \end{aligned}$$

[30 marks]

Marking Scheme

| Item | Mark |
|--|------|
| Comment | 3 |
| Indentation | 2 |
| Read in both value and units | 5 |
| Convert unit in 1 st ext. func. | 10 |
| Calculate area in 2 nd ext. func. | 4 |

| | |
|--|---|
| Calculate volume in 3 rd ext. func. | 4 |
| Display in main() | 2 |

```

Input the radius (>0) in the unit of m, cm, or mm:
120 cm

input radius: 120.00 cm

Input the height (>0) in the unit of m, cm, or mm: 3000 mm

input height: 3000.00 mm

Converted radius: 1.200000 m
Converted volume: 3.000000 m
Area = 4.52 m2
Volume = 13.56 m3

Process returned 0 (0x0)   execution time : 16.663 s
Press any key to continue.
_

```

```

//EEEE1044 final exam 21-22
//Q6 cylindar area and volume

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

float ConvertUnit( float original, char *orunit );
float CalculateArea( float rinm );
float CalculateVolume( float hinm, float areainm2 );

```

```

int main()

```

```

{
    float radius = 0, height = 0;
    float area, rdinm, htinm;
    char runit[3], hunit[3];

```

```

    // read in radius and check value
    while( radius <= 0 )
    {
        printf("Input the radius (>0) in the unit of m, cm, or mm: ");
        scanf("%f %s", &radius, runit );
    } // end while - correct radius input
    printf("\ninput radius: %.2f %s\n\n", radius, runit );

```

```

    // read in height and check value
    while( height <= 0 )
    {
        printf("Input the height (>0) in the unit of m, cm, or mm: ");
        scanf("%f %s", &height, hunit );
    } // end while - correct height input
    printf("\ninput height: %.2f %s\n\n", height, hunit );

```

```

    rdinm = ConvertUnit( radius, runit );
    htinm = ConvertUnit( height, hunit );
    // to check if ConvertUnit works properly
    printf("\nConverted radius: %f m\nConverted volume: %f m\n", rdinm, htinm );

```

```

    area = CalculateArea( rdinm );
    printf("Area = %.2f m2\nVolume = %.2f m3\n", area, CalculateVolume( htinm, area ) );

```

```

    return 0;
} // end main

// to convert all units to m
float ConvertUnit( float original, char *orunit )
{
    char stdunit[3][3] = { "m", "cm", "mm" };
    if ( !( strcmp( orunit, stdunit[0] ) ) )
        return original;
    if ( !( strcmp( orunit, stdunit[1] ) ) )
        return ( original / 100 );
    if ( !( strcmp( orunit, stdunit[2] ) ) )
        return ( original / 1000 );
} // end ConvertUnit

// to calculate area
float CalculateArea( float rinm )
{
    return ( 3.14 * rinm * rinm );
} // end CalculateArea

// to calculate volume
float CalculateVolume( float hinm, float areainm2 )
{
    return ( hinm * areainm2 );
} // end CalculateVolume

```