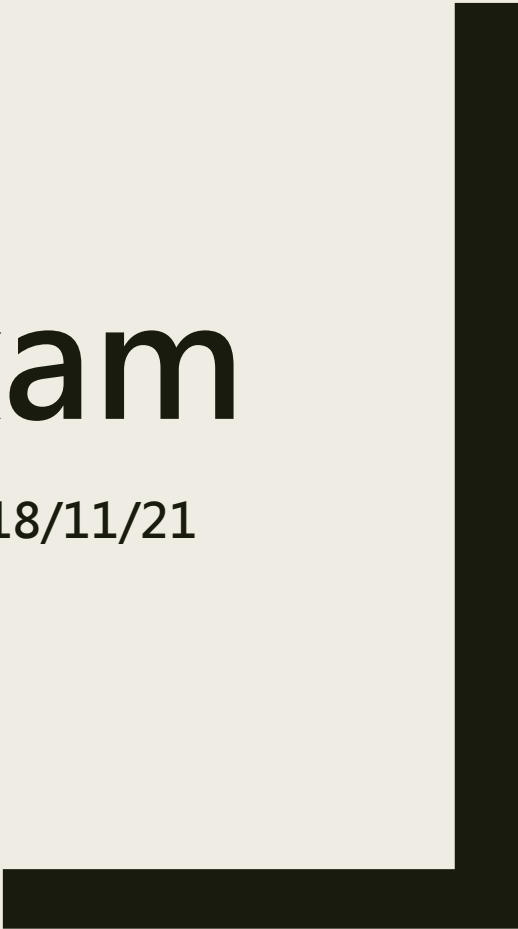# Midterm exam

2018/11/21

考試規則

1. 可以翻閱你覺得有幫助的書、講義

2. 不得作弊，違者依校規論處。

3. 總共只有一次繳交機會，
   請務必確認格式正確後，再舉手找助教繳交。

4. 請將檔案命名為"學號-題號"
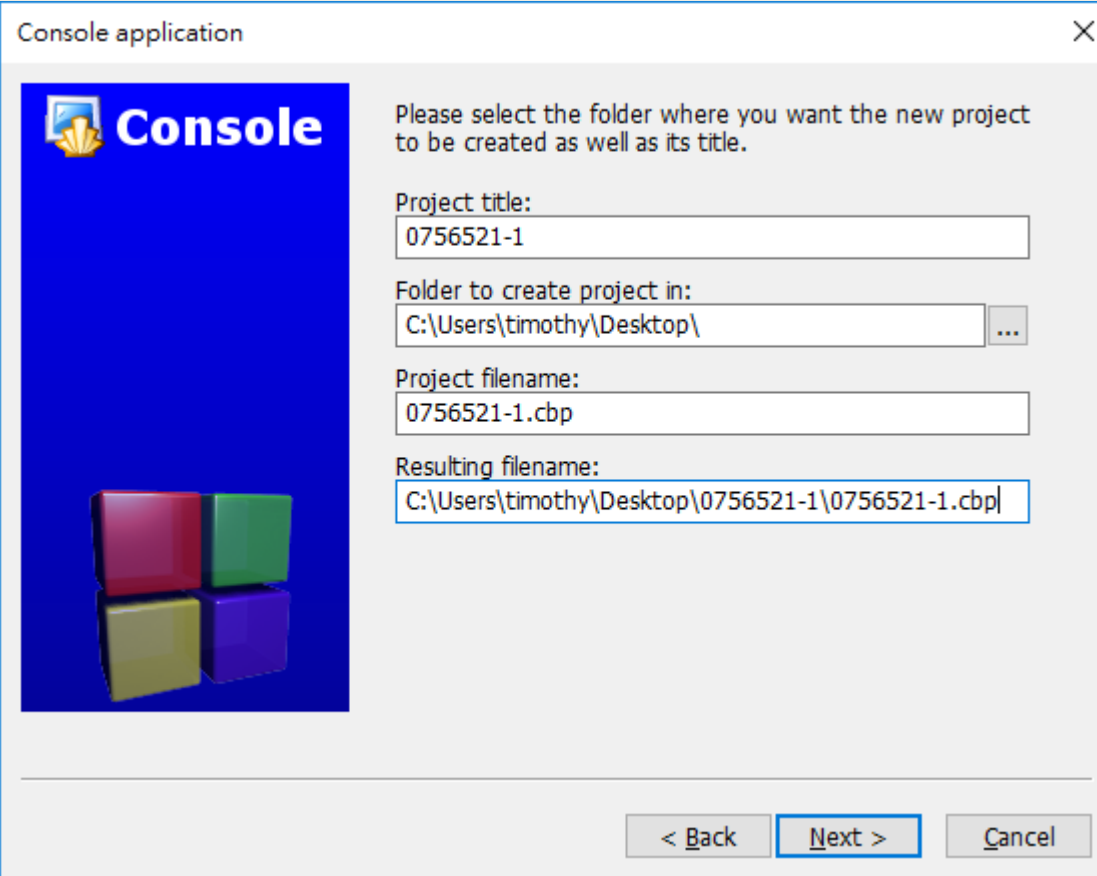   如 0756521-1

格式錯誤

1. **本次考試 Output 中最後一行皆不用換行。**
2. **本次考試 Output 中無空格符號，**

   數字與數字之間改用底線符號 _ 代替。
3. 若有格式錯誤的情形，會將該題分數 ×0.8 計算

計分方式

1. 每一題 25 分，共有 5 題，
   若總分超過 100 分，則以 100 分計算。
2. 時間只有 3.5 個小時 (18:30 ~ 22:00)，
   請同學妥善分配時間進行作答。

File Naming Rule

Name your project file with the student ID and the question number
Eg: Question 1 → 0756521-1
     Question 2 → 0756521-2

Question 1 – Calculate *e*

Description

The value of the mathematical constant *e* can be expressed as an infinite series:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots$$

Write a program that approximates *e* by adding terms until the current term becomes less than ε, where ε is a small (foating-point) number entered by the user. (You should not add the term less than ε in calculation.)

PS: Please define your floating-point variable with float type.

Question 1 – Calculate *e*

Input

The input is ε, which is a floating-number between 0 and 1.

Output

The output is *e*, which is a floating-number calculated by your program.

Question 1 – Calculate *e*

Sample Input 1

0.4

Sample Output 1

2.500000

Sample Input 2

0.01

Sample Output 2

2.708333

Question 2 – Newton's Method

Description

Write a program that use Newton's method to compute the square root of a positive floating-point number.

Let x be the number entered by the user. Newton's method requires an initial guess y for the square root of x (we'll use y=1). Successive guesses are found by computing the average of y and x/y. The following table shows how the square root of 3 would be found:

x	y	x/y	Average of y and x/y
3	1	3	2
3	2	1.5	1.75
3	1.75	1.71429	1.73214
3	1.73214	1.73196	1.73205
3	1.73205	1.73205	1.73205

Question 2 – Newton's Method

Note that the value of y get progressively closer to the true square root of x. For greater accuracy, your program should use variables of type double rather than float. Have the program terminate when the absolute value of the difference between the old value of y and the new value of y is less than the product of 0.00001 and y.

Hint

Call the "fabs" function to find the absolute value of a double. (You'll need to include the <math.h> header at the beginning of your program in order to use "fabs" function.)

Question 2 – Newton's Method

Input

The input consists of a number between 0 and 65535.

Output

Print each step of Newton's method for guessing the square root. The output consists of n lines, where n is the total number of rounds in Newton's method for guessing the square root. Each line shows the variable "x", "y", "x/y" and "the average of y and x/y" with five digits after the decimal point sequentially and separated with "_" symbol.

Question 2 – Newton's Method

Sample Input 1

3

Sample Output 1

3.00000_1.00000_3.00000_2.00000
3.00000_2.00000_1.50000_1.75000
3.00000_1.75000_1.71429_1.73214
3.00000_1.73214_1.73196_1.73205
3.00000_1.73205_1.73205_1.73205

Sample Input 2

8.7

Sample Output 2

8.70000_1.00000_8.70000_4.85000
8.70000_4.85000_1.79381_3.32191
8.70000_3.32191_2.61898_2.97044
8.70000_2.97044_2.92886_2.94965
8.70000_2.94965_2.94950_2.94958
8.70000_2.94958_2.94958_2.94958

Question 3 – Bangla Numbers

Description

Bangla numbers normally use 'kuti' (10000000), 'lakh' (100000), 'hajar' (1000), 'shata' (100) while expanding and converting to text. You are going to write a program to convert a given number to text with them.

Hint

There is an efficient way to implement Bangla Numbers with recursive function. It's worth a try!

Question 3 – Bangla Numbers

Input

The input is the Bangla number you need to calculate, the number is less than 999999999999999.

Output

You have to output a line with the converted text.

Question 3 – Bangla Numbers

Sample Input 1

23764

Sample Output 1

23_hajar_7_shata_64

Sample Input 2

45897458973958

Sample Output 2

45_lakh_89_hajar_7_shata_45_kuti_89_lakh_73_hajar_9_shata_58

Question 4 – Gomoku Validation

Description

Gomoku(五子棋), also called Five in a Row, is an interesting board game. The winner in this game is the first player to form an unbroken chain of five stones horizontally, vertically, or diagonally.

There is a m×n matrix to record Gomoku. If the element is 0, it represents that there are no stone in this place. If the element is 1, it represents that there is a black stone in this place. If the element is 2, it represents that there is a white stone in this space.

Please write a program to find where the middle stone of winning chain is.

PS: There is only one chain in the matrix.

Question 4 – Gomoku Validation

Input

First line of input contains m and n, two numbers are separated by a single space. ($5 \leq m \leq 20$, $5 \leq n \leq 20$). After the first line, there are m lines to containing each row i in the matrix. Each row contains exactly n elements separated by a single space. j-th number in row i is the element M_{ij} of matrix you have to process. ($0 \leq M_{ij} \leq 2$)

Question 4 – Gomoku Validation

Output

The output consists of two line. The first line shows the number of row where the middle stone of winning chain is. The second line shows the number of column where the middle stone of winning chain is.

Question 4 – Gomoku Validation

Sample Input 1

5 5
1 0 0 0 2
0 1 0 2 0
0 0 1 0 0
0 2 0 1 0
2 0 0 0 1

Sample Output 2

3
3

Sample Input 2

6 7
1 0 0 0 2 0 0
0 2 1 0 2 1 0
0 0 2 1 2 0 0
0 1 0 2 2 0 1
0 0 2 2 2 1 0
0 1 0 0 1 1 0

Sample Output 2

3
5

Question 5 – Poker game

Description

There is a poker game with two players, each player can get five cards in the game. Please write a program to judge who wins the game by their poker hand ranking.

The rule of the game is illustrated below: Two players compare their poker hand ranking. If one player has higher poker hand ranking, the player wins this game. If two player have same poker hand ranking, the game is in a tie.

Question 5 – Poker game

Poker hand ranking (from highest to lowest)

1. Royal flush *(同花大順)*
A, K, Q, J, 10, all in the same suit.

2. Straight flush *(同花順)* [*]
Five cards in a sequence,
all in the same suit.

3. Four of a kind *(鐵支)*
All four cards of the same rank.

4. Full house *(葫蘆)*
Three of a kind with a pair.

Question 5 – Poker game

Poker hand ranking (from highest to lowest) (cont.)

5. Flush (同花)
Any five cards of the same suit,
but not in a sequence.

6. Straight (順子) [*]
Five cards in a sequence,
but not of the same suit.

7. Three of a kind (三條)
Three cards of the same rank.

8. Two pair (兩對)
Two different pairs.

Question 5 – Poker game

Poker hand ranking (from highest to lowest) (cont.)

9. Pair (一對)
Two cards of the same rank.

10. High card (高牌)
Simple value of the card.

[*] In our rule, (10, J, Q, K, A), (J, Q, K, A, 2), (Q,K,A,2,3), (K, A, 2, 3, 4) is in a sequence,

Question 5 – Poker game

Input

The input consists of two lines. The first line shows five cards of player A. The second line shows five cards of player B. We will use numbers from 1 to 52 to represent the 52 cards of poker game, the relationship between numbers and cards is illustrated below:

	A	2	3	4	5	6	7	8	9	10	J	Q	K
Hearts (愛心)	1	2	3	4	5	6	7	8	9	10	11	12	13
Tiles (方塊)	14	15	16	17	18	19	20	21	22	23	24	25	26
Clovers (梅花)	27	28	29	30	31	32	33	34	35	36	37	38	39
Pikes (黑桃)	40	41	42	43	44	45	46	47	48	49	50	51	52

Question 5 – Poker game

Output

The output consists of three lines. The first line shows the poker hand of player A. The second line shows the poker hand of player B. The third line shows who wins the game or the game is in a tie.

The format of output is illustrated below:

First line and second line		Third line
Royal flush	Straight	Player A
Straight flush	Three of a kind	Player B
Four of a kind	Two pair	Tie
Full house	Pair	
Flush	High card	

Question 5 – Poker game

Hint

Try to pack your code into function, this way can help your code not to be messy.

For example:
int straight(…): is the poker hand (royal) straight?
bool flush(…): is the poker hand flush?
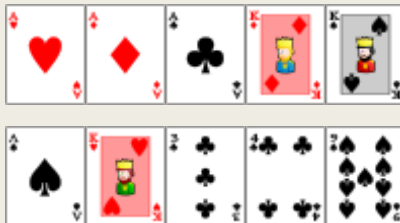int pair(…): how many pairs is in the poker hand?

Question 5 – Poker game

Sample Input 1

1 14 27 26 52

40 13 29 30 48

Sample Output 1

Full house

High card

Player A

Sample Input 2

13 12 24 49 9

46 19 5 43 29

Sample Output 2

Straight

Straight

Tie