Name _____ Student id _____

EN 811 300 Fundamentals of Computer Programming

Final Examination

Faculty of Engineering, Khon Kaen University

Academic Year 2561 Semester 2
22 March 2019, 9:00 – 12:00

Instructions:

1. There are 20 problems. Full scores require every problem solved.
2. This is a closed book exam.
   *** No other reading materials are allowed.**
3. **Network communication is allowed only for submitting the MTE programs to pc2 system.**
   *** Personal communication, social media, file sharing, or internet searching is NOT allowed.**
4. Comment file heading with docstring with student's name, id, and the problem. (This is to double check the submission.)

```
"""
Goodname Happyfamily
621018899-1
P1
"""
```

5. Name the file as follows:
   * Name your submission program by the corresponding problem: **Px.py**
   For example, P1.py for problem 1. P2.py for problem 2, and so on.
6. Submit the program through pc2 system.

==========

Name _____ Student id _____

P1. Write a program to print out "This is a good day."

Example

```
==========
This is a good day.
==========
```

P2. Write a program to ask a user for an input and report it back.

Example

```
==========
How do you feel? fantastic
Ah, you feel fantastic
==========
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P3. Write a program to ask a user for a number, add 1 to it, and report it back.

Example

```
==========
What is your favorite number? 9
I will give you 10
==========
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P4. Write a program to ask a user for his/her weight (in kg), convert it to weights in English unit (lbs) and in Thai unit (Chang), and report them back.

Note: 1 Chang = 1.2 kg; 1 kg = 2.2 lbs.

Name _____ Student id _____

Use P4 Template (`P4Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Example

```
==================================================
What is your weight (in kg)? 102
Your weight is 85.00 chang or 224.40 lbs
==================================================
```

The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P5. Write a program to ask a user for his/her height (in cm), convert it to feet (ft) and inches (in) and report them back.

Note 2.54 cm = 1 in; 12 in = 1 ft.

Use P5 Template (`P5FtInRoundTemplate.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Example

```
==================================================
How tall are you (in cm)? 192
That is around 6 ft 4 in.
==================================================
```

The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P6. Write a program to ask a user for his/her sleeping duration (in hour), if it is less than 7, say "You need more rest."

Say "Thank you" before finishing the program.

Use Sleeping Template (`SleepingTemplate.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Examples

```
==========================================================
How many hours do you sleep a night (in hours)? 8
Thank you.
==========================================================
How many hours do you sleep a night (in hours)? 5
You need more rest.
Thank you.
==========================================================
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P7. Write a program to ask a user for his/her sleeping duration (in hour), if it is less than 7, say "You need more rest." Otherwise, say "Good for you." And, say "Thank you" before finishing the program. Use Sleeping Template (`SleepingTemplate.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Examples

```
==========================================================
How many hours do you sleep a night (in hours)? 8
Good for you.
Thank you.
==========================================================
How many hours do you sleep a night (in hours)? 5
You need more rest.
Thank you.
==========================================================
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P8. Write a program to ask a user for his/her sleeping duration (in hour), if it is less than 7, say "You need more rest."

If it is between 7 and 9, say “Good for you.”
If it is more than 9, say “Don’t you think it’s too much?”
And, say “Thank you” before finishing the program.
Use Sleeping Template (`SleepingTemplate.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Examples

```
==========
How many hours do you sleep a night (in hours)? 8
Good for you.
Thank you.
==========
How many hours do you sleep a night (in hours)? 5
You need more rest.
Thank you.
==========
How many hours do you sleep a night (in hours)? 15
Don't You think it's too much?
Thank you.
==========
```

The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P9. Write a program to ask a user for a number and print out the count from 1 to that number.

Examples

```
==========
Enter a number: 5
1
2
3
4
```

```
5
================================================
Enter a number: 3
1
2
3
================================================
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P10. Write a program to compute the summation: ask a user for a number of values, then ask for each value (as integer), calculate summation, and print out the summation.

Example

```
================================================
Enter a number of values: 5
value:1
value:2
value:3
value:4
value:5
Summation = 15
================================================
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P11. Write a program to compute the summation: keep asking a user for each value (as integer), calculate summation until user enters `0`, and then print out the summation.

Example

```
================================================
value:10
```

```
value:3
value:-4
value:0
Summation = 9
==================================================
```

P12. It is approximated that a man may need to consume food around 2,500 kcal a day, while a woman may need 2,000kcal a day. A boiled egg (~55g) gives around 145 kcal. An apple gives around 35 kcal. A banana gives around 80 kcal. A mandarin orange gives around 23 kcal. A 100-g roasted duck gives around 340 kcal.

Write a program to estimate a healthy daily food consumption: ask a user for sex (Male/Female), then keep asking for his/her food and its estimated kcal until it reaches his/her requirement (M = 2500 kcal and F = 2000 kcal)

Use P12 Template (`P12Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Hint: it is like a summation problem. We don't do anything with the food name. Just provide something that a user can type in.

Examples

```
==================================================
Are you male(M) or female(F)?M
Food:fried rice
est. kcal:350
Food:egg
est. kcal:290
Food:banana
est. kcal:160
Food:orange
est. kcal:69
Food:apple
est. kcal:70
Food:rice w/ 2 sides
```

```
est. kcal:420
Food:sticky rice
est. kcal:700
Food:roasted duck
est. kcal:1000
That is 3059
That would be it for your daily diet.
Be moderate on your diet. Stay healthy.
==========
Are you male(M) or female(F)?F
Food:fried rice
est. kcal:350
Food:egg
est. kcal:290
Food:banana
est. kcal:160
Food:orange
est. kcal:69
Food:apple
est. kcal:70
Food:rice w/ 2 sides
est. kcal:420
Food:sticky rice
est. kcal:700
That is 2059
That would be it for your daily diet.
Be moderate on your diet. Stay healthy.
==========
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

P13. Write a program to calculate a maximum current through a machine. Given a machine operating at 220 volt, ask a user for its

maximum power consumption, calculate its maximum current, and report the calculation.
Note: (max current) = (max power)/(operating volt).
Use P13 Template (`P13Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Example

```
================================================
Max power:100
At 220 V, max current = 0.4545 A
================================================
```

P14. Write a program to calculate an energy stored in a battery. Given a battery operating at 12 volt, ask a user for its capacity (in mAh), calculate its energy stored (in Wh), and report the calculation.
Note: 1 Ah = 1000 mAh; and
(Energy in Wh) = (capacity in Ah) * (operating volt in V).

Use P14 Template (`P14Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)

Example

```
================================================
Battery capacity (mAh):3200
At its 12 V rating, it stores 38.40 Wh.
================================================
```

The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P15. Write a program to calculate a lasting duration of a mobile phone battery. Ask a user for battery energy storage (in Wh), the phone average power consumptions (in mW) in idle and in normal

operation, then calculate estimate duration time the battery will last in each mode, and report the results.
Note: 1 W = 1000 mW;
(Energy Wh) = (Average Power in W) * (Operating time in hr)

Use P15 Template (`P15Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)
Example

```
==================================================
Battery capacity (Wh):35
average power (idle, mW):200
average power (normal, mW):2000
The battery will last 175 hrs in idle mode.
The battery will last 17.5 hrs in normal mode.
==================================================
```

The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P16. Write a program to calculate an energy storage (in kilo calories, kcal) to Joules (J), and Watt-hour (Wh).
Note 1Wh = 3600 J; 1 kcal = 4184 J;

Use P16 Template (`P16Template.py`; note the template is only to ensure the exact display format and allows smooth auto-grading.)
Example

```
==================================================
Energy( in kcal):200
That is 836800.00 J or 232.44 Wh.
==================================================
```

Name _____ Student id _____

P17. Write a function named `sigmoid` taking an argument `a` and return the sigmoid output,

$$sigmoid(a) = \frac{1}{1 + e^{-a}}$$

Hint: e^x can be computed by `math.exp(x)`;

Use P17 template file (`P17SigmoidTemplate.py`; note the template file is only to ensure the exact display format, enforce a proper use of function, and allows smooth auto-grading.)

Example

```
==========================================================
a:0.4
sigmoid(0.40)=0.5987
==========================================================
```

P18. Write a function named `relu` taking an argument `a` and return the relu output,

$$relu(a) = \begin{cases} a & when\ a > 0 \\ 0 & when\ a \leq 0 \end{cases}.$$

Hint: This is a conditional operation.

Use P18 template file (`P18ReLuTemplate.py`; note the template file is only to ensure the exact display format, enforce a proper use of function, and allows smooth auto-grading.)

Example

```
==========================================================
a:-9
relu(-9.00)=0.0000
==========================================================
a:8
relu(8.00)=8.0000
==========================================================
```

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.

Name _____ Student id _____

P19. Write a program to calculate a geometric mean. Geometric mean is calculated as

$$gmean = \sqrt[N]{x_1 \cdot x_2 \cdot x_3 \cdots x_N}.$$

For example,

geometric mean of 3, 4, 8 is

$$\sqrt[3]{3 \cdot 4 \cdot 8} = \sqrt[3]{96} \approx 4.578857.$$

geometric mean of 10, 0.2, 100, 5, 6 is

$$\sqrt[5]{10 \cdot 0.2 \cdot 100 \cdot 5 \cdot 6} = \sqrt[5]{6000} \approx 5.69679.$$

The program asks a user for a number of values to take in and then ask for each value, calculate the geometric mean, and report the calculation. Use P19 template (`P19GMeanTemplate.py`; note the template file is only to ensure the exact display format and allows smooth auto-grading.) Hint: $\sqrt[5]{6000}$ can be computed by `6000**(1/5)`.

Example

```
==================================================
How many values?5
value:10
value:0.2
value:100
value:5
value:6
Geometric mean = 5.696791
==================================================
```

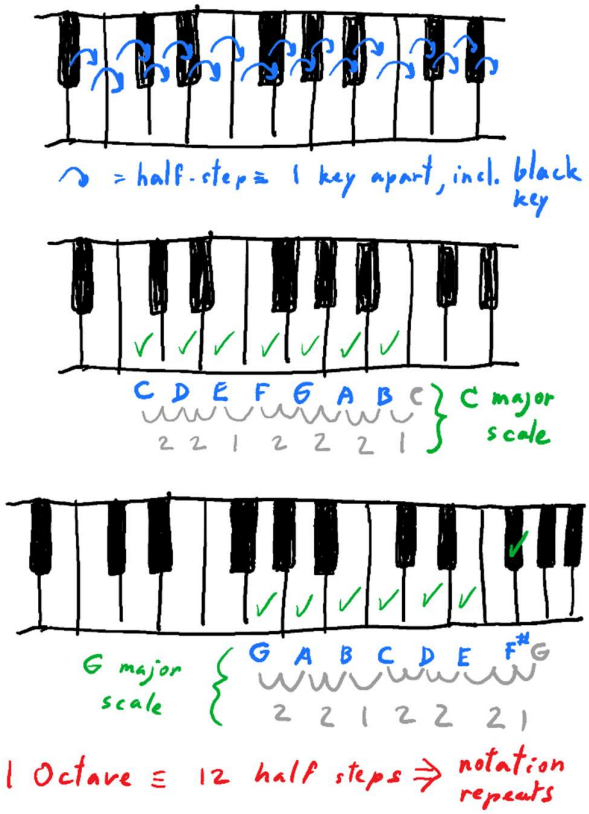
The **`bold font`** represents a user input. The *`italic font`* represents what corresponds to the user input.

P20. Diatonic notes are 7 musical notes on a scale. These 7 notes are defined differently on different scales. For example, the major scale in the key of C has C, D, E, F, G, A, and B. The major scale of G has G, A, B, C, D, E, and F#. The major scale defines its diatonic as follows:

Name _____ Student id _____

Diatonic notes	Example in the key of C
1st note is the scale key	C
2nd: 2 half-steps above the 1st	D (2 keys up)
3rd: 2 half-steps above the 2nd	E (2 keys up)
4th: 1 half-step above the 3rd	F (1 key up: no black key in between E and F)
5th: 2 half-steps above the 4th	G (2 keys up)
6th: 2 half-steps above the 5th	A (2 keys up)
7th: 2 half-steps above the 6th	B (2 keys up)
1st of the next octave: 1 half-step above the 7th	C (1 key up: no black key in between B and C)

Note: half-step is one piano key apart, including a black key.



Write a function named `diatonic` taking an argument `scale_key` (as an integer) and return all 7 notes of the major scale of that scale key (all as integers). Noted that diatonic function works on integers, but the auxiliary file will handle a notation: 1 = C, 2 = C#, 3 = D, and so on.

Use P20 template and auxiliary files (`P20aDiatonicTemplate.py` and `P20aux.py`; note the template and aux files are only to ensure the nice and exact display format, enforce a proper use of function, and allows smooth auto-grading.)

Hint: Recall the triad exercise question.

Examples

==========

Enter the key [1-12]:**1**
Diatonic notes in the key of *C* (*1*)
C D E F G A B

1 3 5 6 8 10 12

==========

Enter the key [1-12]:**6**
Diatonic notes in the key of *F* (*6*)
F G A Bb C D E
6 8 10 11 1 3 5

==========

Enter the key [1-12]:**8**
Diatonic notes in the key of *G* (*8*)
G A B C D E F#
8 10 12 1 3 5 7

==========

Enter the key [1-12]:**11**
Diatonic notes in the key of *Bb* (*11*)
Bb C D Eb F G A
11 1 3 4 6 8 10

==========

The **bold font** represents a user input. The *italic font* represents what corresponds to the user input.