Name ______________________________ Student id _________________

EN 811 300 Fundamentals of Computer Programming

Mid-Term Examination

Faculty of Engineering, Khon Kaen University

Academic Year 2562 Semester 1
27 September 2019, 5:00pm – 8:00pm

Instructions:

1. There are 20 problems. Full scores require every problem solved.
2. This is a closed book exam.
   *** Dictionary is allowed. No other reading materials are allowed.**
3. **Network communication is allowed only for submission of the answers to the designated 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
623049999-9
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. Put `verify.txt` with the other answer files. The `verify.txt` has the content as handed out by exam staff.
7. Write a main program under

```
if __name__ == '__main__':
```

8. Submit the program through the designated system.
   ==

Name ______________________________ Student id _________________

P1. Write a program to ask a user for an input and report it back.
Example

```
=============================================
What are you? an engineer
Ah, you are an engineer
=============================================
```

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

P2. Write a program to ask a user for a number, multiply 2 to it, and report it back.
Example

```
=============================================
What is your favorite number? 9
When double, it becomes 18
=============================================
```

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

P3. Write a function named `exchange`. The function takes 2 arguments: `money` (as float) and `xrate` (as float). Argument `money` represents an amount of money to be exchanged. Argument `xrate` represents an exchange rate to a new currency. For example, we want to exchange 1000 baht to indian rupees. The current exchange rate is 1 baht for 2.34 rupee. Therefore, when invoked a function as follows:

```
=============================================================
rupee = exchange(1000, 2.34)
print(rupee)
=============================================================
```

We will see

```
=========================================================
2340.0
=========================================================
```

P4. Using an ATM abroad will get extra charges. Golden Piggy charges 100B for each transaction abroad and it adds 2.5% (called risk handling) to an amount withdrawn. For example, at 1 baht for 2.34 rupee, if we gets 2340 rupee from ATM, the bank will charge us 100 baht (transaction) + 1000 baht (withdrawn) + 25 baht (2.5%) = 1125 baht.

Write a function named `bloody_atm` to calculate what a customer will be charged for using an ATM abroad. The function takes 4 arguments: amount to withdraw in foreign currency (e.g., rupee), exchange rate (per 1 baht), transaction fee (in baht), and risk handing (in %). Then, the function returns an amount the bank charges us.

For example, when invoked a function as follows:

```
=========================================================
b = bloody_atm(20000, 2.34, 100, 2.5)
print(b)

b = bloody_atm(400000, 870.5, 100, 2.5)
print(b)
=========================================================
```

We will see

```
=========================================================
8860.68376068376
570.9936817920735
=========================================================
```

Name ______________________________ Student id _________________

P5. For simple evaluation of an investment, we may consider investment capital (in baht), estimate annual earnings (in baht), estimate annual cost (in baht), then we want to know a number of years to break even with the investment capital, or simply called Return-Of-Investment (ROI, in years).

Write a function named `roi`. The function takes investment capital (as float), annual earnings (as float), annual cost (as float), then calculate the ROI, round it up to the closed integer, and return the round-up ROI. Note `math.ceil(3.2)` gives `4`. We can calculate roi from

annual profit = annual earnings – annual cost;

roi (in years) = investment capital/annual profit.

For example, if we get an offer of a business required 14,000,000 baht worth of investment, with estimate annual earnings of 4,000,000 baht and annual cost of 1,200,000. Therefore, when invoked by

```
========================================================
r = roi(14000000, 4000000, 1200000)
print(r)
========================================================
```

We will see

```
========================================================
5
========================================================
```

That is, it is expected to take about 5 years to return of the investment capital.

P6. Write a function named `thresholding`. The function takes two arguments: `a` and `tau` and returns 1 if `a` ≥ `tau` or returns 0 otherwise.

For example, when invoked a function as follows:

```
========================================================
b = thresholding(5, 3)
print(b)
```

```
b = thresholding(10, 10)
print(b)

b = thresholding(2, 5)
print(b)
========================================================
```
We will see
```
========================================================
1
1
0
========================================================
```

P7. Write a function named `safe_log`. The function takes a floating-point number as its argument `a`. It returns `log(a)` for `a` > 0 and string “`-Inf`” for `a` = 0, and string “`NaN`” for `a` < 0.
Hint: `math.log(x)` returns logarithm of x.

For example, when invoked a function as follows:
```
========================================================
b = safe_log(200)
print(b)

b = safe_log(0)
print(b)

b = safe_log(-1)
print(b)
========================================================
```
We will see
```
========================================================
5.298317366548036
-Inf
NaN
========================================================
```

Name ______________________________ Student id _________________

P8. Given a formulation

$$s = \sum_{i=0}^{n} e^{-i},$$

write a function named `sum_exp` taking an integer `n` and returning summation `s`.
Hint: `math.exp(x)` returns e^x.

For example, when invoked a function as follows:

```
========================================================
r = sum_exp(0)
print(r)

r = sum_exp(1)
print(r)

r = sum_exp(10)
print(r)

r = sum_exp(100)
print(r)
========================================================
```

We will see

```
========================================================
1.0
1.3678794411714423
1.581950285167711
1.5819767068693267
========================================================
```

P9. Write a program to keep ask an integer number from a user. Add up all even numbers the user enters. Ignore any odd number. End the program when the user enters a zero or a negative number. Print out the summation.
Hint: even numbers are 2, 4, 6, 8, ...; odd numbers are 1, 3, 5, 7, ...

Name ______________________________ Student id _________________

Example 1

```
=========================================================
Enter a number: 1
Enter a number: 2
Enter a number: 3
Enter a number: 4
Enter a number: 5
Enter a number: 0
All even numbers are summed up to 6
=========================================================
```

Example 2

```
=========================================================
Enter a number: 0
All even numbers are summed up to 0
=========================================================
```

Example 3

```
=========================================================
Enter a number: 2
Enter a number: 8
Enter a number: 9
Enter a number: 7
Enter a number: 2
Enter a number: 11
Enter a number: 4
Enter a number: 0
All even numbers are summed up to 16
=========================================================
```

Example 4

```
=========================================================
Enter a number: 3
Enter a number: -2
All even numbers are summed up to 0
=========================================================
```

Name ______________________________ Student id _________________

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

P10. Write a function named `engr_prefix`. The function takes a floating-point number `x` and returns a number (as a floating-point number) and a prefix (as a string). To simplify this task for our limited time, the function uses only 3 prefixes:

Prefix	**x**
M	1,000,000 ≤ |x|
k	1000 ≤ |x| < 1,000,000
(no prefix)	1≤ |x| < 1,000
m	|x| < 1

Hint: statement `return 3.45, "M"` *will return 2 values: 3.45 as float and "M" as string.*

For example, when invoked a function as follows:

```
========================================================
r = engr_prefix(450)
print(r)

r = engr_prefix(3450)
print(r)

r = engr_prefix(8200000)
print(r)

r = engr_prefix(0.75)
print(r)

r = engr_prefix(0.0145)
print(r)
========================================================
```

We will see

```
========================================================
(450, '')
```

```
(3.45, 'k')
(8.2, 'M')
(750.0, 'm')
(14.5, 'm')
========================================================
```