Student ID : _____

CITY UNIVERSITY OF HONG KONG

Course code & title : CS1302 Introduction to Computer Programming

Session : Semester B 2021-22

Time allowed : One (1) Hours

This paper has **9** pages (including this page).

Note:

1. Answer ALL the questions in the space provided within each question.

Question	Part 1 (5 marks)	Part 2 (15 marks)	Part 3 (5 marks)	Part 4 (15 marks)	Total (40 marks)
Marks					

2. Do NOT remove the staple or separate the paper.
3. This question paper should NOT be taken away.

Remarks:

For all written code required by the questions:

1. You should give precise Python code with proper programming styles, in particular, appropriate code design, naming and code formatting. Marks may be deducted for redundant or unnecessary code.
2. Unless specifically mentioned, modules/libraries/functions other than built-in functions, math modules are not allowed.
3. Please write down your answer on your A4 or single line paper.

Note that this is just a sample midterm exam paper. It doesn't represent all the question types/numbers/difficulty of the real midterm exam.

---

*This is a **closed-book** examination.*

No materials or aids are allowed during the whole examination. If any unauthorized materials or aids are found on a candidate during the examination, the candidate will be subject to disciplinary action.

Part 1 (5 marks)

Multiple choice questions (1 mark for each question).
Note that there may be more than 1 correct option for each question, and you must select all the correct options to get the full mark. There is no partial mark such as 0.5. The Python code is in brown color.

(1) Given variables n and x which contain positive non-zero integer values, x/n == x//n is always True when:

A. x is divisible by n
B. x > n
C. x < n
D. x == n

Correct answer: A D

(2) Which of the following is correct concerning a < x < b in Python code, where x, a, and b are integers?

A. It is the same as the expression (a<x) < b.
B. It must be False if a is larger than b
C. It is the same as the expression a<x and a < b
D. If a = 3, x = True, b = 8, then a<x<b will return False

Correct answer: B, D

(3) Given variables a, b which contain two different integer values. What will be done if we execute the following statement two times?

```
a, b = b, a
```

A. The values of a and b will become the same integer value
B. a and b will have the original values
C. a and b will become None
D. The code will raise an error if executed in Python

Correct answer: B

(4) Consider the code:

```
if (status != "Full") and (input("Want something to eat? [y/n] ") == "y"):
    print("eat")
```

The output will NOT say "eat" when

A. status is not "Full" and no matter what the user inputs
B. status is "Full"
C. the user inputs "y"
D. status is not "Full" and the user inputs "n"

Correct answer: B, D

(5) print(isclose(a,b)) and print(a==b) may output different results when

A. a or b is a floating point value
B. a and b are integers and their values are not the same
C. a and b are integers and their values are the same
D. a is Boolean True and b is floating point number 21.3

Correct answer: A

Part 2 (15 marks)

For each of the following code, select its correct type if it is a python expression, and select the option "Has no values" otherwise.
The result should be one of the followings.

- None Type
- str
- Has no values
- int
- function
- float
- bool

Python code	Result
0 and '1'	int
None	NoneType
not 1	bool
i += 1	Has no values
print(1)	NoneType
1.0	float
x = 10	Has no values
123 // 10 % 10	int
True	bool
1e10	float
if input(): 1	Has no values
input	function
1 if input() else 0	int
3 * '3'	str
'inf'	str

Part 3 (5 marks)

1. Please explain the difference of for loop and while loop (2 marks).

   Answer: Main differences between “for” and “while” loop
   - The for statement iterates through a collection or iterable object or generator function. The while statement simply loops until a condition is False.
   - For loop is used when we are aware of the number iterations at the time of execution. on the other hand, in “while” loop, we can execute it even if we are not aware of the number of iterations. Or in other words, for loop is a definite loop which has a definite number of iterations before the execution of the loop. while statement is indefinite loop where the number of iterations is unknown before the execution of the loop.

2. Please explain the difference of pass, break and continue statement (3 marks).

   Answer:
   Pass： In Python, pass is a null statement. The interpreter does not ignore a pass statement, but nothing happens and the statement results into no operation. The pass statement is useful when you don’t write the implementation of a function but you want to implement it in the future.

   Break: The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.

   Continue: The continue statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

Part 4 (15 marks)

Please complete the following programming questions (3 marks for each question)

1. We are to put n oranges into big boxes and small boxes. We first fill them into big boxes. The remaining that are not enough to fill a big box will be put in small boxes.

E.g., Given box capacities 8 and 5 respectively. To hold 10 oranges, we put 8 in a big box and the remaining 2 in a small box.

Complete the function, count_boxes(n, big_size, small_size), that returns the counts of boxes required for n oranges with box capacities big_size , and small_size. Assume that n, big_size, small_size are positive non-zero integers.

For example:

Test	Result
bg, sm = count_boxes(31, 8, 5) print(bg, "big box(es)", sm, "small box(es)")	3 big box(es) 2 small box(es)
bg, sm = count_boxes(36, 12, 5) print(bg, "big box(es)", sm, "small box(es)")	3 big box(es) 0 small box(es)
bg, sm = count_boxes(92, 20, 4) print(bg, "big box(es)", sm, "small box(es)")	4 big box(es) 3 small box(es)

Your Answer:

```
def count_boxes(n, big_size, small_size):
    # Your code below




    return bg, sm


Sample answer:

def count_boxes(n, big_size, small_size):
    bg = n//big_size
    n -= bg*big_size
    sm = n//small_size
    n -= sm*small_size
    if n:
        sm+=1
    return bg, sm
```

2. Complete the function **convert_h24_to_h12(h24)** that prints the conversion of **h24**, an integer for an hour number in the 24-hour format, to the12-hour format. Assume 0 <= h24 <= 24.

Note: - Mind the specific numbering and labels for the hours at noon and midnight.
- Your code should **NOT** contain more than six **if** or **elif** clauses

For example:

Test	Results
convert_h24_to_h12(9)	9:00 => 9:00 a.m.
convert_h24_to_h12(20)	20:00 => 8:00 p.m.
for h in range(25): convert_h24_to_h12(h)	0:00 => 12:00 a.m. 1:00 => 1:00 a.m. 2:00 => 2:00 a.m. 3:00 => 3:00 a.m. 4:00 => 4:00 a.m. 5:00 => 5:00 a.m. 6:00 => 6:00 a.m. 7:00 => 7:00 a.m. 8:00 => 8:00 a.m. 9:00 => 9:00 a.m. 10:00 => 10:00 a.m. 11:00 => 11:00 a.m. 12:00 => 12:00 noon 13:00 => 1:00 p.m. 14:00 => 2:00 p.m. 15:00 => 3:00 p.m. 16:00 => 4:00 p.m. 17:00 => 5:00 p.m. 18:00 => 6:00 p.m. 19:00 => 7:00 p.m. 20:00 => 8:00 p.m. 21:00 => 9:00 p.m. 22:00 => 10:00 p.m. 23:00 => 11:00 p.m. 24:00 => 12:00 a.m.

Your Answer:

```
def convert_h24_to_h12(h24):
    # Add your code below to decide the values for h12 and label


   print(str(h24) + ":00 => " + str(h12) + ":00 " + label)


Sample answer:
def convert_h24_to_h12(h24):
    if h24==0:
        h12 = 12
        label = "a.m."
    elif h24==12:
        h12 = 12
        label = "noon"
    elif h24==24:
        h12 = 12
        label = "a.m."
    else:
        h12 = h24%12
        if h24<12:
            label="a.m."
        else:
            label="p.m."
    print(str(h24) + ":00 => "+str(h12) + ":00 " + label)
```

3. Write a Python function **print_number(a, b, n)**, which iterates the integers from **0** to **n** and print them out. For multiples of **a**, print "CS" instead of the number; for multiples of **b**, print "1302". For numbers that are multiples of both **a** and **b** print "CS1302".

Assume that **a**, **b**, and **n** are non-zero positive numbers. Also, note that **0** is a multiple of any numbers.

For example:

Test	Result
print_number(3,5,7)	CS1302 1 2 CS 4 1302 CS 7

Answer:

```
# Complete the code below

def print_number( a, b, n):




Sample answer:

def print_number(a,b,n):
    for i in range(n+1):
        if i % a == 0 and i % b == 0:
            print("CS1302")
        elif i % a == 0:
            print("CS")
        elif i % b == 0:
            print("1302")
        else:
            print(i)
```

4. Consider the Fibonacci number of order **n**:

$$F(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & n \geq 2 \end{cases}$$

Your task: Finish the function **isFibonacci(f)** that takes in a positive integer **f** and returns **True** if **f** is a Fibonacci number, and **False** otherwise.
Note:
- You should make use of the given function, **fibonacci_iteration**, which is already provided in the answer box.
- You may assume **f**<50000. Therefore, the running speed should not be a concern unless your logic makes substantially redundant operations.

For example:

Test	Results
print(isFibonacci(144))	True
print(isFibonacci(337))	False
print(isFibonacci(377))	True

```
def fibonacci_iteration(n): # return the Fibonacci number of order n
    if n > 1:
        x, y = 0, 1
        while n > 1:
            x, y, n = y, x + y, n - 1
        return y
    elif n == 1:
        return 1
    else:
        return 0


def isFibonacci(f):
# Complete your code below


Sample answer:

def isFibonacci(f):
    i = 0
    while fibonacci_iteration(i)<f:
        i+=1
    return fibonacci_iteration(i)==f
```

5. You are given the **is_vowel(c)** function that returns **True** if the character **c** is a vowel ('a', 'e', 'i', 'o', or 'u'), and **False** otherwise.

Your tasks:
(1) Write a function **vowel_part(str)** that takes in a string of word **str** and returns a copy of the word with non-vowel letters replaced by the underscore character "_". E.g., vowel_part("learning") should return "_ea__i__"
(2) Write a function **match** that takes two input words and prints their vowel parts. If their vowel parts are the same, then print it once only. Assume that the two words have the same length.
Note: In this question we only deal with lowercase letters.

For example:

Test	Results
match("ink","ice")	i__ i_e
match("hobby","throw")	_o___ ___o_
match("happy","sandy")	_a___
match("learning", "teaching")	_ea__i__

```
# The given is_vowel function
def is_vowel(c):
    return c in "aeiou"


# Please complete the function below
def vowel_part(str):
    result=""

    return result

# Please complete the function below
def match(       ):



Sample answer:
def vowel_part(str):
    result=""
    for c in str:
        if is_vowel(c):
            result=result+c
        else:
            result=result+"_"
    return result

def match(s1, s2):
    v1 = vowel_part(s1)
    v2 = vowel_part(s2)

    if v1==v2:
        print(v1)
    else:
        print(v1)
        print(v2)
```

-- END OF PAPER --