University of Saskatchewan
Department of Computer Science
CMPT 141.3
PRACTICE Final Examination
Timeless

Marks: 75 **Time: 180 minutes**

Name: _____ **NSID:** _____ **Student #:** _____

Instructions

- Don't Panic
- This is a PRACTICE final exam. It has not been, nor will ever be, used as the actual CMPT 141 final exam
- But it has been carefully constructed to be as similar as possible to the actual exam
- On the real exam, you will use OpScan (bubble) sheets for the multiple choice. Take this into account if timing yourself
- This exam has written questions which you will answer in this exam booklet.
  - Write the answers in the space provided. Use the back of the page if you need extra space (or for rough work).
- The mark value of each question is provided in the left margin.
- You may bring a single 8.5x11 sheet of hand-written notes into the exam
- Read every question carefully!

Academic Honesty

This exam is an individual undertaking. Cheating on an exam is considered a serious offence by the University and can be met with disciplinary action, including suspension or expulsion. By handing in this exam and op-scan sheet you affirm that this work is entirely your own.

It is considered an academic offence to take this examination paper from the exam room.

NSID: _____

Part I — Multiple Choice

Choose the best answer for each question. Choose only one answer per question.

(1) 1. Which of the following is an example of an algorithm?

A. A voting ballot listing candidates and their political parties

B. A box of parts for building a desk

C. A recipe for baking a cake

D. A movie theater schedule listing showtimes

E. A highway road map

For questions 2 through 4, assume that the following variable initializations are given.

```
x = 3.3
y = 2.2
z = 1.1
a = 3
b = 2
c = 5
p = 1.5 + c * b
q = x - a // c + z
r = x + a * b / c - y
```

(1) 2. What is the value of `p` after the above initializations?

A. 13.0 B. 13 C. 11.5 D. 11.0 E. 11

(1) 3. What is the value of `q` after the above initializations?

A. 1.16 B. 0.049 C. 3.8 D. 3.4 E. 4.4

(1) 4. What is the value of `r` after the above initializations?

A. 0.0 B. 3.4 C. 1.1 D. 2.3 E. 1.42

(1) 5. In total, how many **function calls** can be found in this program?

```
name1 = "Ash"
name 2 = "Gary"
longest = max(len(name1), len(name2))
print("Longest name has", longest, "letters.")
```

A. 0 B. 1 C. 2 D. 4 E. 6

(1) 6. What is displayed on the console by this program?

```
a = 7
float(a)
print(a)
```

A. a B. "a" C. 7 D. 7.0 E. "7.0"

(1) 7. Consider the following four versions of the `formatTime()` function:

```
def formatTime(totalMinutes):
    h = totalMinutes // 60
    m = totalMinutes % 60
    return str(h) + ":" + str(m)
```

(Version 1)

```
def formatTime(totalMinutes):
    h = str(totalMinutes // 60)
    m =str(totalMinutes0 % 60)
    print( h + ":" + m )
```

(Version 2)

```
def formatTime(totalMinutes):
    totalMinutes = input()
    h = totalMinutes // 60
    m = totalMinutes % 60
    return str(h) + ":" + str(m)
```

(Version 3)

```
def formatTime(totalMinutes):
    h = str(totalMinutes // 60)
    m = str(totalMinutes % 60)
    return print(h + ":" + m)
```

(Version 4)

If you ran the code:

```
print(formatTime(95))
```

which version of the function would exactly cause the text `1:35` to be displayed?

A. Version 1 B. Version 2 C. Version 3 D. Version 4

(1) 8. Which line of code would need to replace the BLANK in order to print `Hello, Pikachu!` to the console?

```
BLANK
    message = "Hello, " + s + "!"
    print(message)

greet("Pikachu")
```

A. `def greet(str(s)):`

B. `def greet():`

C. `def greet("s"):`

D. `def greet(input(s)):`

E. `def greet(s):`

(1) 9. Which line of code would need to replace the BLANK in order to print `1234` to the console?

```
def combine_digits(d1, d2):
    answer = int( str(d1) + str(d2) )
    return answer

BLANK
print(answer)
```

A. `combine_digits(1000, 234)`

B. `combine_digits(str(12), str(34))`

C. `answer = combine_digits(12, 34)`

D. `combine_digits(int(12), int(34))`

E. `combine_digits("12", "34")`

(1) 10. Recall that `isdigit()` is a **string object method** that **returns** `True` if its string object contains only digits and `False` otherwise. Which expression needs to replace the BLANK in order for the code to convert the string `s` to an integer whenever `s` consists only of digits?

```
if BLANK:
    num = int(s)
```

A. `isdigit(s)`

B. `s = s.isdigit()`

C. `s.isdigit()`

D. `isdigit(str(s))`

E. `s == isdigit()`

(1) 11. Which line of code correctly imports the `math` library using the name `m`?

A. `import math as m`

B. `import math as math`

C. `import m as math`

D. `import math`

E. `import m`

For questions 12 through 13, assume that `s` is defined as follows:

```
s = "DetectivePikachu"
```

(1) 12. What is the value of the expression: `s[-1:-len(s)-1:-1]` ?

A. `"uhcakiPevitceteD"` B. `"DetectivePikachu"` C. `"u"` D. `"uh"` E. `""`

(1) 13. What is the value of the expression: `s[1:len(s):4]` ?

A. `"etPc"` B. `"Dete"` C. `"achu"` D. `"uhca"` E. `"eia"`

Assume that the following variable initializations are given for questions 14 through 15.

```
x = 4
y = 77
```

(1) 14. Which expression evaluates to **TRUE**?

A. `x > y or x > 0`

B. `x == 4 and y > 100`

C. `not x == 4`

D. `not (x > 0 and y > 0)`

E. `x > y or y == 42`

(1) 15. Which expression evaluates to **TRUE**?

A. `not (x == y and x > 10 or y > 10)`

B. `x > 0 and x > y and y > 0`

C. `x > 0 and (x == y or y < 100)`

D. `not x > 0 or not y > 0`

E. `(x > 0 or y > 0) and (x > y or y == 100)`

(1) 16. What is displayed to the console by this program?

```
speed = 45
SPEED_LIMIT = 50
turbo = True

if speed < SPEED_LIMIT:
    speed = speed + 10
elif speed < SPEED_LIMIT and turbo:
    speed = speed + 25

print(speed)
```

A. 45 B. 50 C. 55 D. 70 E. 80

(1) 17. What is displayed to the console by this program?

```
ptype = "grass"
level = 3

if ptype == "electric":
    print("pikachu")
elif ptype == "grass":
    if level == 1:
        print("bulbasaur")
    elif level == 2:
        print("ivysaur")
else:
    print("unknown")
```

A. `pikachu` B. `bulbasaur` C. `ivysaur` D. `unknown` E. Nothing is displayed

(1) 18. Consider the following programs. Which of them displays `11` to the console?

```
i = 0
while i < 11:
    i = i + 1
print(i)
```

(Program 1)

```
for i in range(0, 11, 2):
    pass
print(i)
```

(Program 2)

```
i = 11
while i <= 11:
    i = i + 1
print(i)
```

(Program 3)

```
i = 1
while i < 12:
  i = i + 1
print(i)
```

(Program 4)

A. Program 1 B. Program 2 C. Program 3 D. Program 4

(1) 19. Which line of code needs to replace the BLANK in order for the program below to print out the integers from 0 to 3, each on their own line?

```
N = 4
BLANK
    print(number)
```

A. `for number in len(N):`

B. `for number in range(len(N)):`

C. `for N in range(number):`

D. `for number in range(N):`

E. `for number in N:`

(1) 20. What is output to the console by the code snippet below?

```
total = 0
for i in range(0, 2):
    for j in range(0, 2):
        for k in range(0, 3):
            total = total + 1

print(total)
```

A. 4 B. 6 C. 8 D. 9 E. 12

(1) 21. What is displayed to the console by the program below?

```
total = 0
i = 0
j = 0
while i < 4:
    while j < 6:
        total = total + 1
        j = j + 1
    i = i + 1
print(total)
```

A. 1 B. 4 C. 6 D. 24 E. Nothing - the code has an infinite loop

(1) 22. What is displayed to the console by this program?

```
L = [10, 20, 30, 40]
for i in range(len(L)):
    L[i] = 3
print(L)
```

A. [13, 23, 33, 43]

B. [3, 20, 30, 40]

C. [10, 20, 30, 40]

D. [3, 3, 3, 3]

E. [10, 20, 30, 43]

(1) 23. What is displayed to the console by this program?

```
nums = [2, 1, 0]
names = ["Ada", "Babbage", "Cookie"]
print(names[nums[2]])
```

A. 2 B. 1 C. 0 D. Ada E. Babbage F. Cookie

(1) 24. What is displayed to the console by this program?

```
L = []
for i in range(4):
    L.append([i])
print(L)
```

A. []

B. [0, 1, 2, 3]

C. [[0, 1, 2, 3]]

D. [[0], [1], [2], [3]]

E. [[0, [1, [2, [3]]]]]

(1) 25. What is displayed to the console by this program?

```
arr = ["A, "B", "C", "D", "E"]
n = arr[0]
c = arr[-1]

for i in range(len(arr)):
    arr[i] = c
    c = n
    p = i + 1
    if (p >= len(arr)):
        p = 0
    n = arr[p];

print(arr[0])
```

A. A B. B C. C D. D E. E

(1) 26. Which **list comprehension** results in the list `[0, 1, 2, 3, 4, 5]`?

A. `[item in range(len(range(6)))]`

B. `[item for item in range(6)]`

C. `[item for list in range(6)]`

D. `[item in range(len(6))]`

E. `[item for range(6)]`

(1) 27. Assume `L` refers to the list `[10, -5, 15, -20, 30]`. Which **list comprehension** would result in the list `[10, 15, 30]`?

A. `[x for x in L if x > 0]`

B. `[x if x > 0 for L]`

C. `[x in L if x > 0]`

D. `[x in range(len(L)) if x > 0]`

E. `[x in range(L) for x > 0]`

(1) 28. Assume `L` refers to the list `[ [10, 20, 30], [40, 50, 60]`. Which **list comprehension** would result in the list `[ [10, 20], [40, 50] ]`?

A. `[[x[0], x[1]] for x in L]`

B. `[x.pop(-1) for x in L]`

C. `[x.remove(-1) for x in L]`

D. `[x[i] for i in range(len(L))]`

E. `[list.append(x) for x[0] in L]`

(1) 29. Observe the dictionary, `ptypes`. What is displayed to the console by this program?

```
ptypes = {"fire" : 2,
          "water": 3,
          "grass": 1}
print( ptypes[1] + ptypes[2])
```

A. `2` B. `3` C. `4` D. `5` E. Nothing - the code will raise an error

(1) 30. Which expression should replace the BLANK so that this program displays `66000000`?

```
countries = { "canada" : {"name" : "canada", "population" : 37000000},
          "france" : {"name" : "france", "population" : 66000000},
          "iran" : {"name" : "iran", "popuplation" : 81000000} }
print(BLANK)
```

A. `countries[0]["population"]`

B. `countries[1]["population"]`

C. `countries["france"]["population"]`

D. `countries["france"][1]`

E. `countries["population"][0]`

(1) 31. Which of the following statements would **open** the file `data.txt` for **reading**?

A. `open(data)`

B. `open(data.txt, w)`

C. `open("data.txt", "r")`

D. `open(str(data), "r")`

E. `open(data.txt, "rw")`

(1) 32. Suppose that a data file called `measurements.txt` looks like this:

```
5
7
12
9
6
```

Presuming the file has already been opened for reading, which line would need to replace the BLANK in the following code in order to print `39` (the sum of all of the numbers in the file) to the console?

```
sum = 0
for line in f:
    BLANK
    sum = sum + m

print(sum)
```

A. `d = line.strip()`

B. `d = line.rstrip().split()`

C. `d = int(line.split()`

D. `d = int(line)`

E. `d = line[0]`

(1) 33. Which of these is an example of a **recursive function**?

```
def FunctionOne(N):
    if N <= 0:
        return 0
    else:
        return pow(2, N-1)
```

```
def FunctionTwo(N):
    result = 1
    for i in range(N):
        result = result * 2
    return result
```

```
def FunctionThree(N):
    return max(N, max(N, 0))
```

```
def FunctionFour(N):
    if N <= 1:
        return 1
    return 2*FunctionFour(N-1)
```

A. FunctionOne B. FunctionTwo C. FunctionThree D. FunctionFour

(1) 34. The following function is intended to compute the factorial of a number, N. Which line needs to replace the BLANK for the function to work correctly?

```
def fact(N):
    if N <= 1:
        return 1
    else:
        return BLANK
```

A. `N * fact(N-1)`

B. `N + fact(N-1)`

C. `fact(N) + fact(N)`

D. `N * (N-1)`

E. `N + N^2`

(1) 35. The function `getNegs()` is supposed to return a list containing only the negative numbers from a list of integers. For the purpose of **testing**, what is the **expected** result of this function for the test case `[10, 0, -3, 6, -7]`?

```
def getNegs(L):
    result = []
    for num in L:
        if not num > 0:
            result.append(num)
    return result
```

A. `[-3, -7]` B. `[0, -3, -7]` C. `[]` D. `True` E. `False`

(1) 36. The function `toNumber()` is supposed to convert a string that represents a single number to its matching numerical value, returning either a float or integer (as appropriate). Which of the following code snippets properly implements a **test case** for this function?

```
input = "hello"
if toNumber(input) != False:
    print("Fault detected for: ", input)
    print("input is not a valid a number")
```

Test 1

```
input = "7"
if toNumber(input) == 7:
    print("Fault detected for input", input)
    print("input is a single digit")
```

Test 2

```
input = "12.34"
if not (type(toNumber(input))) is float:
    print("Fault detected for input", input)
    print("returned data type was not float")
```

Test 3

```
input = "-123"
if toNumber(input) != -123:
    print("Fault detected for: ", input)
    print("input is multiple digits that begin with minus sign")
```

Test 4

A. Test 1 B. Test 2 C. Test 3 D. Test 4

(1) 37. The function `validName()` is supposed to return `True` if its string parameter `v` is a valid name for a Python variable, and `False` otherwise. However, it contains an error; in particular, the test case `pika4ever` gives `False`, when it should give `True`. Which single-line substitution below is enough to fix the code so that it works as intended?

```
1  def validName(v):
2      answer = False
3      if len(v) > 0:
4          answer = True
5      for i in range(len(v)):
6          char = v[i]
7          if char.isdigit():
8              answer = False
9          elif not (char == "_" or char.isalnum()):
10             answer = False
11
12     return answer
```

A. change line 3 to `if len(v) > 1:`

B. change line 7 to `if char.isdigit() and i == 0:`

C. change line 8 to `answer = True`

D. change line 9 to `elif not (char == "_" and char.isalnum()):`

E. change line 10 to `answer = True`

(1) 38. How many data items would **linear search** examine when searching for the value `15` in the list `[10, 50, 40, 70, 30, 80, 20, 60]`?

A. 0 B. 1 C. 3 D. 7 E. 8

(1) 39. How many data items would **binary search** examine when searching for the value `15` in the list `[10, 20, 30, 40, 50, 60, 70, 80]`? Recall that, when needed, binary search rounds indices down.

A. 0 B. 1 C. 3 D. 7 E. 8

(1) 40. Consider sorting the list `[30, 20, 10, 50, 40]` in ascending order using **insertion** sort. What is the **first** number inserted into *S* (the sequence of sorted items that insertion sort builds up)?

A. 10 B. 20 C. 30 D. 40 E. 50

(1) 41. Consider sorting the list `[10, 40, 20, 30]` in ascending order using **merge** sort. What is the result of the first **divide** step?

A. `[10]` and `[40, 20, 30]`

B. `[10]` and `[20, 30, 40]`

C. `[10, 40]` and `[20, 30]`

D. `[10, 40, 20]` and `[30]`

E. `[10, 20]` and `[30, 40]`

(1) 42. Consider sorting the list `[10, 50, 30, 40, 20, 30]` in ascending order using **quick** sort. Using `30` as the **pivot**, what is the result of the first **divide** step?

A. `[10, 50],[30, 30],[40, 20]`

B. `[10, 20],[30] [40, 50]`

C. `[10, 20],[30, 30],[50, 40]`

D. `[10, 20],[30],[30, 40, 50]`

E. `[10, 50,[30],[40, 20, 30]`

(1) 43. Recall that the **merge** algorithm used by merge sort combines two separate lists together, which we will call `L1` and `L2`. This merge algorithm **assumes** that:

A. `L1` and `L1` are the same length

B. everything in `L1` is smaller than everything in `L2`

C. `L1` is shorter than `L2`

D. `L1` and `L2` are each sorted

E. `L1` is empty and `L2` is non-empty

(1) 44. What is the value of the decimal number 117 in binary?

A. 0b001111011

B. 0b011100010

C. 0b010110111

D. 0b01110101

E. 0b10010111

(1) 45. What is the result of the binary addition 0b01011010 + 0b00101101?

A. 0b101101011

B. 0b010000111

C. 0b011010001

D. 0b101010101

E. 0b101111111

(1) 46. In the **von Neumann** computer architecture, computer programs are stored in the computer's:

A. arithmetic-logic unit (ALU)

B. central processing unit (CPU)

C. main memory

D. instruction register (IR)

E. machine cycle

Part II — Written Answers.

Answer each question in the space provided on this question paper.

(3) 47. A weather station measures the temperature and the windspeed. On very cold days, the station needs to issue a frostbite warning.

Write a function to deterine whether a frostbite warning should be issued. The function should accept **2 parameters**: temperature and windspeed. If the temperature is less than -40, then a warning is needed. But if the temperature is less than -25 and the windspeed is at least 20, a warning is also needed.

The function should not print anything, but should **return** `True` if a frostbite warning is needed and `False` otherwise.

(2) 48. Write a program that **uses a loop** to calculate and display to the console the sum of all the even numbers from 1 up to (and including) 1000.

(3) 49. Write a program that **uses loops** to print out all 2-letter permutations of the string `ABCDEFG`. Recall that for permutations, order matters: so `AF` is a different permutation from `FA`.

(1) 50. Supplose `classlist` is a list of student names. Write a statement that adds the name `Ash` to the end of `classlist`.

(2) 51. Suppose `classlist` is a list of student names. Use a **list comprehension** to create a new list of all of the names from `classlist` that start with the letter `A`.

(3) 52. Suppose `classgrades` is a **dictionary** that maps student names to their grade in a class. The student names are the keys, and the grades are the values.

Write a program that prints out the name of each student with a grade of 80 or higher.

(3) 53. Suppose `book.txt` is a plain text file filled with text from a book. An excerpt from this file might look like this:

```
One fish.  Two fish.
Red fish.  Blue fish.
```

Write a program that reads this file and displays to the console **how many words** are in the book. You may assume that words on the same line are separated by spaces. For the sample above, your program would report that there are 8 words.

(2) 54. Write a **recursive function** that accepts a non-negative integer `N` as a parameter and returns the value of `2` to the power of `N`. For this question you **cannot use loops** in any way; you must use recursion instead.

(3) 55. Write a **recursive function** that searches for a specific character in a string. Your function must use **exactly two** parameters: `C`, which is the character we want to find, and `S`, which is the string that we're searching. The function should return `True` if `C` can be found anywhere in `S`, and `False` otherwise. For this question, you are **not allowed** to use loops or the Python `in` operator.

(4) 56. Obvserve the **header and docstring** for the function `merge()`:

```
def merge(L1, L2):
    """
    Returns a sorted list that is the combination of L1 and L2.
    L1, L2: Both lists.  Assumed to be individually sorted
    return: A sorted list consisting of all items in L1 and L2
    """
```

Create two **black box** test cases for `merge()`.

Use only **valid inputs** for the function as inputs for your test cases. (i.e. don't bother with test cases that violate the function's assumptions about the input)

Input:

Output:

Reason:

Input:

Output:

Reason:

(3) 57. Detective Pikachu is working on a case. To solve it, he will have to walk through the city and collect clues.

The city is an `NxM` grid represented by a 2D list (i.e. a list-of-lists where each sublist is the same length) of strings. If a space on the grid contains a `"*"` (asterisk) character, then that space contains a clue. If it does not, then there is no clue in that space.

Write a function that will compute and **return** the number of clues Pikachu will find from his walk through the city. The parameters to your function must be:

- `city`: a 2D list of strings
- `x` and `y`: integer coordinate pair that gives location of Pikachu's apartment, his starting point for his walk
- `path`: a string of any length that will only contain the characters `N`, `S`, `E`, and `W` that represents Pikachu's path that he walks

The four letters in the path stand for **North**, **South**, **East** and **West**. These are the directions that Pikachu will move, starting from his apartment. Pikachu collects a clue whenever it is in the same space as him.

You are guaranteed that the path will not take Pikachu out of the city, so you do not need to check for that case. But he may reverse direction or walk in circles, so you DO need to make sure that he doesn't count the same clue twice!

Example: Pikachu's apartment is at (0, 0) (the top left corner). His path is `"SES"`. This means that Pikachu will walk South from (0, 0), then East, then South again. His final position will be at (2, 1) and he will have found 2 clues.

	0	1	2
0	_	_	*
1	*	_	_
2	_	*	_

Example of a 3x3 city

This description was kind of long, so start your work on the next page.

(Q62 cont...)