

15110 PRINCIPLES OF COMPUTING – SAMPLE EXAM 1

Name _____ Section _____

Andrew id _____

*Directions: Answer each question neatly in the space provided.
Please read each question carefully. You have 50 minutes for
this exam. No electronic devices allowed. Good luck!*

1	_____
2	_____
3	_____
4	_____
5	_____
6	_____
TOTAL	_____

1. History of computation

(a) [4 pts] Match each item in the left column with the most relevant item in the right column.

Jacquard's loom	_____	1. Electromechanical computer
ENIAC	_____	2. Enigma cipher
Moore's Law	_____	3. Polynomial function
Difference Engine	_____	4. Vacuum tubes
Harvard Mark I	_____	5. First programmer
Grace Hopper	_____	6. Hollerith tabulating machine
Ada Lovelace	_____	7. Exponential function
Alan Turing	_____	8. Debugging

(b) [4 pts] A byte is _____ bits, so a kilobyte is _____ bits.

(c) [4 pts] Moore's Law says that the _____ of integrated circuit
chips _____ every 18 months.

2. This problem focuses on expressions and data types.

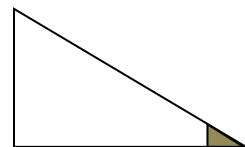
(a) [6 pts] For each of the following Python expression, write down the value that would be output if the expression was evaluated using a python3 interpreter .

40 // 9	_____	15.0 // 2	_____
2 * 2 ** 4	_____	15 % 2	_____
6 + 4 * 2 - 1	_____	2 != 2	_____

(b) [3 pts] Write a Python3 function `triangle_area` that takes two parameters `h` and `b`, respectively, for the height and base of a triangle, and returns the area of the triangle given by the formula

$$A = \frac{1}{2} (\text{height} \times \text{base}).$$

(c) [3 pts] Write a Python function `truncated_triangle` that takes height and base parameters (`h` and `b`) as input, and computes the area of a triangle with the tip cut off. (That is, the white area of the triangle shown below.) The tip is also a triangle; its height and base are 10% of the height and base, respectively, of the larger triangle, as shown in the figure. Use the `triangle_area` function in your solution.



(d) [2 pts]

```
def mystery1(m,n):  
    i = 0  
    while i <= (n-1):  
        i = i + 1  
        print(i ** m, end = " ")
```

The Python function above prints a sequence of numbers. Which of the following is the output of the function expressed in terms of m and n? Circle your answer.

1 2^m 3^m ... n^m OR 1 m² m³ ... mⁿ

(e) [5 pts] If the print statement was taken outside of the while loop to occur right after the while statement, as shown below, what would the function call `mystery2(2, 3)` print?

```
def mystery2(m,n):  
    i = 0  
    while i <= (n-1):  
        i = i + 1  
    print(i ** m, end = " ")
```

(f) (5 pts) Consider the following Python function:

```
def mystery3(m,n):  
    i = 0  
    result = 0  
    while i <= (n-1):  
        i = i + 1  
        result = result + i ** m  
    return result
```

What would the value of the variable x be after executing the following assignment statement below?

```
x = mystery3(2,4)
```

3. This question focuses on the list data type and variable assignments.

(b) [4 pts] Assume the following list definition in Python.

```
a = [1, 2, [3, 4, 5], 6]
```

What would be displayed in a python3 shell for each of the following expressions?

len(a) _____
a[2] _____
a[2][0] _____
a + a _____

(c) [10 pts] Suppose that we type the following assignments in a python3 shell in the given order.

```
x = 5  
y = 10  
x = x + y  
y = y + x
```

For each of the expressions below write down the value that would be output if the expression was evaluated by a python3 interpreter after making the assignments above.

x _____

y _____

(y % x) // x _____

4. This question focuses on looping.

(a) [8 pts] We wish to define a Python3 function `out_of_order` that takes an “almost sorted” list as input and returns the first item that is not in ascending order. The function should return `None` if the list is entirely in ascending order. For example, `out_of_order([1, 5, 17, 12, 24])` should return 12, since 12 is less than the preceding item, 17. Complete the following iterative function `out_of_order`.

```

def out_of_order(list):
    index = _____
    while index < _____ :
        if _____ > _____ :
            return _____
    index = _____
    return _____

```

(b) [8 pts] Consider the following recursive algorithm for returning the first item in a list that is not in ascending order, else None. Complete the recursive definition of `out_of_order`.

1. If the list has fewer than two elements, return None.
2. If the first element in the list is greater than the second element, return the first element.
3. Otherwise return the result of a recursive call on the tail of the list (i.e., everything beyond the first element.)

```

def out_of_order(list):
    if _____ < 2 :
        return _____
    elif _____ > _____ :
        return _____
    else:
        return _____

```

(c) [2 pts] Give an example of a six element list that would be a worst case input for `out_of_order`.

(d) [2 pts] What is the big O worst case complexity of `out_of_order`?

5. This question deals with searching and sorting.

(a) [2 pts]

What is the big O complexity of binary search? _____

What is the big O complexity of insertion sort? _____

(b) [6 pts] Fill in the table below to show how binary search would locate the value "e" in the list ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k"]. Use the binary search algorithm taught in the book and covered in lecture. Note: this table may contain extra rows.

Iteration	Low	High	Mid	list[mid]
1	-1	11	5	f

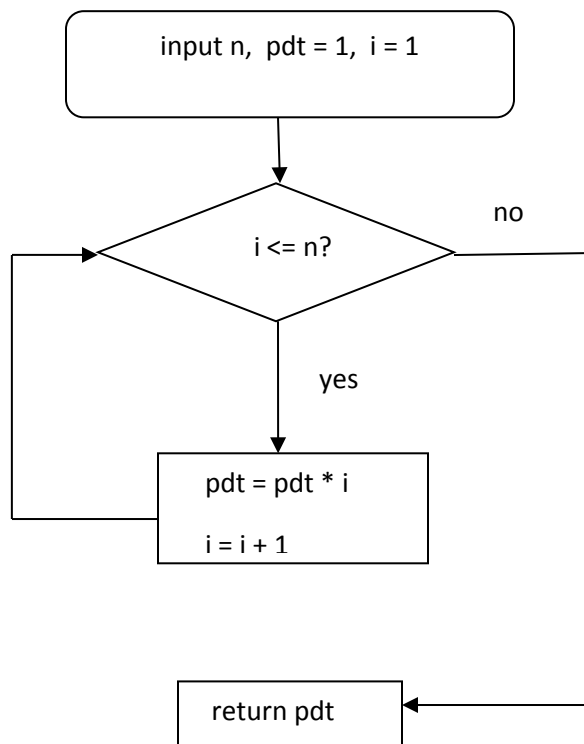
(c) [6 pts] For each sorting algorithm described below, give its correct name:

- For each input item, find its proper position in the result list and add it at that position.

- For each position i in the list, find the index of the smallest item at or to the right of position i , and swap $list[i]$ with $list[index_of_smallest]$.

- Organize the inputs into n groups of size 1. Systematically combine adjacent groups to form $n/2$ sorted groups, each of size 2. Repeat the process, combining adjacent groups of size 2 to form $n/4$ sorted groups of size 4. Keep going until you have one sorted group of size n .

Consider the following flow chart.



d) [6 pts] Convert the above flow chart into Python code. Use the following outline

```
def mystery_function(n):  
    _____  
    _____  
    while ( _____ ):  
        _____  
        _____  
    return _____
```

e) [4 pts] Explain in one sentence the purpose of this code (using n in your answer)

6. This question is based on your readings from the book *Blown to Bits*.

[6 pts] When you print a report using a laser printer, can you assume that no one can tell who printed it? Give a yes/no answer followed by a one sentence explanation.