

Name: _____

Section: _____ Date: _____

INSTRUCTIONS:

- (1) DO NOT OPEN YOUR EXAM BOOKLET UNTIL YOU HAVE BEEN TOLD TO BEGIN.
- (2) This exam booklet contains 24 questions, each of which will be weighted equally at 5 points each. It also contains a 30-point written question. The total points for the exam is 150 points.
- (3) You may use one 8.5" x 11" note sheet during the exam. No other reference materials or electronic devices may be used during the examination, i.e. no calculators, cell phones, mp3 players, etc.. Paper dictionaries are allowed.
- (4) Questions will not be interpreted during the examination.
- (5) You should choose the **single best** alternative for each question, even if you believe that a question is ambiguous or contains a typographic error. If a question has more than one best answer, credit will be given for any of the correct answers. Provide only one answer.
- (6) Please fill in the requested information at the top of this exam booklet.
- (7) Use a #2 pencil to encode answers on the OMR form.
- (8) Please encode the following on the OMR form:
 - Last name and first initial
 - MSU PID
 - Exam form (2 A)
- (9) Please sign the OMR form.
- (10) Only answers recorded on your OMR form for multiple choice questions will be counted for credit. Completely erase any responses on the OMR form that you wish to delete.
- (11) You must turn in this exam booklet, the OMR form, your note sheet, and your scrap paper when you have completed the exam. Be sure your name is on any of these items that you would like to have returned.
- (12) When leaving, please be courteous to those still taking the exam.

1	2	3	4	5	6	7	8	9	10	11	12	13
D	D	C	A	D	C	D	C	A	A	A	C	C
14	15	16	17	18	19	20	21	22	23	24		
A	A	C	B	D	A	A	C	D	B	D		

- (1) What is printed when the following code is executed?

```
L1 = [1, 2, 3]
L2 = [4, 5]
L2.append(L1)
print(L1)
```

- (a) [1, 2, 3, 4, 5] (b) [1, 2, 3, [4, 5]]
(c) [4, 5] (d) [1, 2, 3]
(e) None of (a)–(d)

- (2) What is printed when the following code is executed?

```
L1 = [1, 2, 3]
L2 = [4, 5]
L2.append(L1)
print(L2)
```

- (a) [1, 2, 3, 4, 5] (b) [1, 2, 3, [4, 5]]
(c) [4, 5] (d) [4, 5, [1, 2, 3]]
(e) None of (a)–(d)

- (3) What is printed when the following code is executed?

```
L1 = [1, 2, 3]
L2 = [4, 5]
L2.append(L1)
print(L2[2][2])
```

- (a) [4, 5] (b) [1, 2, 3]
(c) 3 (d) 2
(e) None of (a)–(d)

- (4) What is printed when the following code is executed?

```
print(list("Spartan")[-1])
```

- (a) n (b) natrapS (c) Sparta
(d) ['S', 'p', 'a', 'r', 't', 'a'] (e) None of (a)–(d).

- (5) What is printed when the following code is executed?

```
print(list("Spartan")[:-1])
```

- (a) n (b) natrapS (c) Sparta
(d) ['S', 'p', 'a', 'r', 't', 'a'] (e) None of (a)–(d).

- (6) Which of the following statements about a string method is correct?
- (a) A string method can modify the object it operates on and it can also return a value.
 - (b) A string method cannot modify the object it operates on and it cannot return a value.
 - (c) A string method cannot modify the object it operates on but it can return a value.
 - (d) A string method can modify the object it operates on but it cannot return a value.
 - (e) None of (a)–(d) are correct.

```
l = [c.lower() for c in 'Catch 22']  
print(l)                                # Line 1  
l = [c.lower() for c in 'Catch 22'  
      if 'a' <= c <= 'z']  
print(l)                                # Line 2  
l = [c for c in 'Catch 22'.split()]  
print(l)                                # Line 3
```

FIGURE 1

- (7) In Figure 1, what is printed by Line 1?
- (a) ['Catch', '22']
 - (b) ['catch']
 - (c) ['a', 't', 'c', 'h']
 - (d) ['c', 'a', 't', 'c', 'h', ' ', '2', '2']
 - (e) None of (a)–(d).
- (8) In Figure 1, what is printed by Line 2?
- (a) ['Catch', '22']
 - (b) ['catch']
 - (c) ['a', 't', 'c', 'h']
 - (d) ['c', 'a', 't', 'c', 'h', ' ', '2', '2']
 - (e) None of (a)–(d).
- (9) In Figure 1, what is printed by Line 3?
- (a) ['Catch', '22']
 - (b) ['catch']
 - (c) ['a', 't', 'c', 'h']
 - (d) ['c', 'a', 't', 'c', 'h', ' ', '2', '2']
 - (e) None of (a)–(d).

```
def my_fun(n, m):  
    n += m  
    return n  
  
print(my_fun(6,2.0))           # Line 1  
print(my_fun("!", "Ha"))       # Line 2  
l = list(range(1,3))  
print(my_fun(l,l))             # Line 3  
print(l)                       # Line 4
```

FIGURE 2

(10) In Figure 2, what is printed by Line 1?

- (a) 8.0 (b) 12.0 (c) 8
(d) 12 (e) None of (a)–(d).

(11) In Figure 2, what is printed by Line 2?

- (a) '!Ha' (b) '!Ha!Ha' (c) 'Ha!'
(d) 'Ha!Ha!' (e) None of (a)–(d).

(12) In Figure 2, what is printed by Line 3?

- (a) [1, 2] (b) [1, 2, 3] (c) [1, 2, 1, 2]
(d) [1, 2, 3, 1, 2, 3] (e) None of (a)–(d).

(13) In Figure 2, what is printed by Line 4?

- (a) [1, 2] (b) [1, 2, 3] (c) [1, 2, 1, 2]
(d) [1, 2, 3, 1, 2, 3] (e) None of (a)–(d).

```
def big_i(s_lst):
    try:
        result, remain = s_lst[0], s_lst[1:]
        for s in remain:
            result = result & s
    except IndexError:
        result = set()
    return result

print(big_i([]))                                # Line 1
arg = [{0, 1, 2, 3}, {1, 2, 3, 4}, {2, 3, 4, 5}]
print(len(arg))                                # Line 2
print(arg[1])                                  # Line 3
print(big_i(arg))                              # Line 4
```

FIGURE 3

(14) In Figure 3, what is printed by Line 1?

- (a) `set()` (b) `{}` (c) `[]`
(d) an `IndexException` (e) None of (a)–(d).

(15) In Figure 3, what is printed by Line 2?

- (a) 3 (b) 4 (c) 12
(d) None of (a)–(c).

(16) In Figure 3, what is printed by Line 3?

- (a) `set()` (b) 1 (c) `{1, 2, 3, 4}`
(d) an `IndexException` (e) None of (a)–(d).

(17) In Figure 3, what is printed by Line 4?

- (a) `set()` (b) `{2, 3}` (c) `{0, 1, 2, 3, 4, 5}`
(d) an `IndexException` (e) None of (a)–(d).

```
def init(fname):
    fobj = open(fname, 'r')
    a_dict, ct = {}, 1
    for l in fobj:
        l = l.strip()
        for c in l.lower():
            if c.isalpha():
                a_dict[c] = ct
        ct = ct + 1
    fobj.close()
    return a_dict

data = init("infile.txt")

if 2 in data:                # first if
    print(data[2])
else:
    print("Not there!")

if 'o' in data:              # second if
    print(data['o'])
else:
    print("Not there!")

print(max(data.values()))    # Line 1
print(max(data.items()))     # Line 2
```

infile.txt:

Uno, dos, tres ...

FIGURE 4

- (18) What is printed by the first if-statement in Figure 4 if the contents of `infile.txt` are as shown at the right?
- (a) d (b) o (c) dos
(d) Not there! (e) None of (a)–(d).
- (19) What is printed by the second if-statement in Figure 4 if the contents of `infile.txt` are as shown at the right?
- (a) 2 (b) 3 (c) 8
(d) Not there! (e) None of (a)–(d).

(20) What is printed by Line 1 in Figure 4 if the contents of `infile.txt` are as shown at the right?

- (a) 3 (b) 6 (c) u
(d) uno (e) None of (a)–(d).

(21) What is printed by Line 2 in Figure 4 if the contents of `infile.txt` are as shown at the right?

- (a) 3:'t' (b) (3, 't') (c) ('u', 1)
(d) ('uno', 1) (e) None of (a)–(d).

```
list1 = [1, 2]
list2 = [1, 2, list1]
list3 = list2[:]
print(list1 is list2[2], list1 == list2[2])      # Line 1
print(list1 is list2[:-1], list1 == list2[:-1])  # Line 2
print(list2[2] is list3[2], list2[2] == list3[2]) # Line 3
```

FIGURE 5

(22) What is printed by Line 1 in Figure 5?

- (a) False, False (b) False, True (c) True, False
(d) True, True (e) None of (a)–(d).

(23) What is printed by Line 2 in Figure 5?

- (a) False, False (b) False, True (c) True, False
(d) True, True (e) None of (a)–(d).

(24) What is printed by Line 3 in Figure 5?

- (a) False, False (b) False, True (c) True, False
(d) True, True (e) None of (a)–(d).

Written: (30 points) You will write two functions, `get_index` and `print_index`, to use with the following main function.

```
import string
# your function definitions go here
def main():
    universe = ["woodchuck", "would", "could",
                "much", "how", "wood", "banana"]
    fobj = open('twister.txt', 'r')
    index = get_index(fobj, universe)
    print_index(index)
```

This main function prints an index of the words that appear in `universe` (a list of strings) and also in the tongue twister in `twister.txt`, indicating the line numbers at which a word appears. To illustrate, assume the file is as shown below:

```
How much wood would a woodchuck chuck,
If a woodchuck could chuck wood?
He would chuck, he would, as much as he could,
And chuck as much as a woodchuck would
If a woodchuck could chuck wood.
```

Then calling `main()` should produce the following output:

could	2, 3, 5
how	1
much	1, 3, 4
wood	1, 2, 5
woodchuck	1, 2, 4, 5
would	1, 3, 4

In more detail, write correct functions that meet the following specifications:

(1) `get_index(fobj, words)`

- Inputs:
 - `fobj`: a file object, attached to a file that is open for reading
 - `words`: a list of words (strings)
- Returns:
 - a dictionary with the words as keys and the set of line numbers containing a word as the word's value
- Has no side effects.

(2) `print_index(index)`

- Inputs:
 - `index`: a dictionary of the form returned by `get_index`
- Returns: nothing (None)
- As a side effect, prints the index with
 - one word to a line, in alphabetical order;
 - line numbers separated by commas and in ascending order;
 - first line number of each entry lined up as shown.

(3) Ignore capitalization. (Convert all words to lowercase.)

(4) Ignore punctuation at the beginning and ending of words. (Do not worry about embedded punctuation.)

(5) Do not use global variables.

Work out your answer first on a scratch paper. Then copy all of your answer *neatly* on the sheet provided. Be careful to show indentation carefully.

This page left intentionally blank.

Name AND Section: _____