

CPTS 111 — FINAL EXAM

Closed Book/Three Sheets of Notes Allowed
No electronic devices of any kind!

Directions:

1. Breathe in deeply, exhale slowly, and relax.
2. No hats or sunglasses may be worn during the exam.
3. Do not look at anyone else's exam or let anyone else look at yours!
4. Be neat! If we can't read what you've written, we can't give you credit!
5. If an answer is a `float`, make it clear by using a decimal point.
6. If an answer is an integer, make it clear by *not* using a decimal point.
7. If an answer should be in quotes, then use quotes.
8. For multiple choice questions, **circle your answer**. Only one answer is correct.
9. **When writing code, indent clearly when you should indent.**
10. When writing code, do your best to indicate what you can do even if you can't do everything.
11. **When coding, be sure to use the prompts shown in the examples.** *If no prompts are shown, use one that fits the problem!!!*
12. **Each question is worth 2 points unless otherwise indicated.**
13. **Read each question carefully and check your work if you finish early!**
14. **Do not use docstrings in any functions you define.**

1. If we want to use the `matplotlib.pyplot` module to plot a dashed red line with circles, which format string should we use?

- (a) `'rc--'`
- (b) `'ro--'`
- (c) `'rc:-'`
- (d) None of the above.

2. Suppose we have the following list:

```
nums = [13, 4, 2, 7, 16, 11, 5, 32]
```

To add 1 to all the odd values in this list, we must use a counting `for`-loop rather than an iterating `for`-loop.

- (a) True
- (b) False

3. What is the correct value of `cents` for:

```
dollars, cents = divmod(1300, 100)
```

- (a) 1300
- (b) 130
- (c) 13
- (d) 0

4. What is the output of the following code?

```
song = 'Bohemian Rhapsody'  
print(song[2 : 5])
```

- (a) ohem
- (b) ohe
- (c) hemi
- (d) hem

5. Given the dictionary `capitals`:

```
{'Texas': 'Austin', 'Oregon': 'Salem', 'Washington': 'Olympia'}
```

which of the following commands will add a new entry to `capitals`?

- (a) `capitals['Idaho'] = 'Boise'`
- (b) `capitals[Idaho] = 'Boise'`
- (c) `capitals.append(Idaho = 'Boise')`
- (d) None of the above.

6. Given:

```
>>> def limits(height, weight):  
    if (48 < height) and (85 <= weight <= 185):  
        print('Limits met')  
    else:  
        print('Limits not met')  
    return
```

Which of the following function calls will *NOT* result in Limits met?

- (a) `limits(60, 105)`
 - (b) `limits(72, 185)`
 - (c) `limits(48, 85)`
 - (d) None of the above.
7. What is the output of the following code?

```
phrase = 'We good?'  
phrase[0] = 'B'  
print(phrase)
```

- (a) Be good?
 - (b) We good?
 - (c) B
 - (d) None of the above.
8. What is the output of the following code?

```
x = 5  
x = x // 4  
x += 1
```

- (a) 2
 - (b) 3
 - (c) 6
 - (d) None of the above.
9. What is the output of the following code?

```
nums = {1: [1], 2: [2, 2], 3: [3, 3]}  
nums[3].append(3)  
print(nums)
```

- (a) `{1: [1], 2: [2, 2], 3: [3, 3, 3]}`
- (b) `{1: [1], 2: [2, 2], 3: [3]}`
- (c) `{1: [1], 2: [2, 2], 3: [3, 3], 3:[3]}`
- (d) None of the above.

10. What is the output of the following code?

```
values = 1, 42
print(values)
```

- (a) 1 42
- (b) 1, 42
- (c) (1, 42)
- (d) None of the above.

11. What is the output of the following code?

```
nums = [42, 1.618, -2, 3.141, 12, -99]
nums.sort(reverse=True)
print(nums)
```

- (a) [-99, -2, 1.618, 3.141, 12, 42]
- (b) [42, 1.618, -2, 3.141, 12, -99]
- (c) [-99, 42, 12, 3.141, -2, 1.618]
- (d) None of the above.

12. Lists, dictionaries, and strings are sequences; lists and dictionaries are mutable, but strings are immutable.

- (a) True
- (b) False

13. What is the output of the following code?

```
>>> fruits = ['pears', 'plums', 'figs']
>>> fruit_str = '* '.join(fruits)
>>> print(fruit_str.split())
```

- (a) pears* plums* figs
- (b) ['pears*', 'plums*', 'figs*']
- (c) ['pears*', 'plums*', 'figs']
- (d) pears*plums*figs*

14. What is the output of the following code?

```
>>> title = 'The Cat in the Hat'
>>> title[6 : 12]
```

- (a) 't in t'
- (b) 'at in t'
- (c) 't in th'
- (d) 'at in'

15. What is the output of the following code?

```
phrase = 'She sells seashells down by the seashore.'  
print(phrase.count('w'))
```

- (a) 0
- (b) 1
- (c) 2
- (d) None of the above.

16. Given the following dictionary, which command do we use to delete a key-value pair?

```
ages = {'Ali': 22, 'Bea': 21, 'Cal': 20}
```

- (a) `ages.remove('Ali')`
- (b) `ages.clear('Ali')`
- (c) `del ages['Ali']`
- (d) `remove ages['Ali']`

17. What is the output of the following code?

```
>>> drink = 'Spumante'  
>>> print(drink[-1])
```

- (a) Spumant
- (b) pumante
- (c) e
- (d) An error message.

18. What is the output of the following code?

```
s = '2, 3, 5, 7, 11'  
print(s.split(','))
```

- (a) `['2', ' 3', ' 5', ' 7', ' 11']`
- (b) `[2, 3, 5, 7, 11]`
- (c) `235711`
- (d) `2 3 5 7 11`

19. What is the output of the following code?

```
>>> print(ord(chr(65)))
```

- (a) A
- (b) 65
- (c) Not enough information given to tell.
- (d) An error message.

20. The standard way for importing the numpy module is `import numpy as np`.

- (a) True
- (b) False

21. What is the output of the following code?

```
print('have yourself a merry little christmas'.title().swapcase())
```

- (a) Have Yourself A Merry Little Christmas
- (b) hAVE yOURSELF a mERRY lITTLE cHRISTMAS
- (c) Have yourself a merry little christmas
- (d) None of the above.

22. The non-void, built-in function `input()` returns a string.

- (a) True
- (b) False

23. Suppose we've imported `matplotlib.pyplot` as `plt`. Which statement below is required in order for a plot to be displayed?

- (a) `plt.plot()`
- (b) `plt.display()`
- (c) `plt.show()`
- (d) None of the above.

24. What is the correct indexing to obtain 99 for the list:

```
nums = [3.141, 42, [0, 1], 1.618, [[2, 3, 5, 7], [33, 66, 99]]]
```

- (a) `nums[5][2][3]`
- (b) `nums[4][1][2]`
- (c) `nums[6][1][2]`
- (d) None of the above.

25. What is the output of the following code?

```
for i in range(1):
    for j in range(5):
        print(i+1, chr(j+65), sep='', end='')
```

- (a) 1A 1B 1C 1D 1E
- (b) 1A 2B 3C 4D 5E
- (c) 1A 2B 3C 4D
- (d) None of the above.

26. [3 pts] Write a single line of code with a single set of quotes to print: `Python rocks!`

27. [3 pts] Write a single line of code to print the email address for Jean given the following dictionary.

```
email = {'Jean': jean@wsu.edu, 'Juan': juan@wsu.edu}
```

28. [3 pts] Write a single line of code to join items in the list `dlist` by `' ; '` and assign the result to the lvalue `desserts`.

29. [3 pts] Write a single line of code to split the string `desserts` at `' ; '` and assign the result to the lvalue `dlist`.

30. [3 pts] Write a single `if` statement that tests to see whether `num` is even, smaller than 100, and larger than or equal to 10. Don't forget the colon!

31. [7 pts] Given the list of lists:

```
desserts = [['pies', 0], ['cakes', 0], ['fruit tarts', 0]]
```

Define a void function `order()` with one parameter `desserts`, the list of lists, which uses a counting `for`-loop to prompt a user for the number of desserts in an order as shown below and updates the order using the input values entered by the user (shown in boldface).

```
Enter number of pies in order: 3
Enter number of cakes in order: 4
Enter number of fruit tarts in order: 8
```

32. [7 pts] Write a void function `main()` that 1) opens an input file `cipher.txt` for reading, assigning it to the lvalue `file_in`, 2) opens an output file `clear.txt` for writing, assigning it to the lvalue `file_out`, 3) uses an iterating `for`-loop to read each line in the input file, 4) strips and passes each line to the function `decrypt()` which returns the decrypted line which is assigned to the lvalue `clear`, 5) *prints* each decrypted line to the output file, and finally 6) closes the files. **Do not define the `decrypt()` function!**

33. [7 pts] Define a non-void function `snowball_fight` with the four parameters `n`, an integer, `width`, the width of the game screen in pixels, `height`, the height of the game screen in pixels, and `dim`, the width or height of a snowball, which returns a list of lists `snowballs` with the positions of `n` snowballs. The positions `x` and `y` for each snowball should be determined randomly using `r.randrange()` with three arguments. Note that if the width of a game screen is 500 pixels, the first pixel occurs at 0 and the last pixel at 499. The same is true for the height.

34. [7 pts] Define a non-void function `odd()` with one parameter `nums`, a list. This function should determine whether a number in `nums` is odd, and if it is, it should add it to a list called `odds`. `odds` should be returned to the calling function.

35. [7 pts] Given a list of lists `players`, e.g.,

```
players = [  
    ['Dazee', 48032, ['Atma', 'Expedition', 'Konami']],  
    ['Queen', 51026, ['Arcana', 'Atma', 'Munchkin']],  
    ['MsWow', 39289, ['Arcana', 'Expedition', 'Konami']],  
    ['KBird', 42271, ['Arcana', 'Atma', 'Konami']]  
]
```

where each inner list consists of a player's name, their overall score, and a list of the RPG card games they play. Define a void function `leaderboard()` with one parameter `players`, a list of lists, which prints a histogram of the scores *only for players who play the game **Atma***, where each asterisk represents approximately 1000 points. For the list above, the output would be:

```
>>> leaderboard(players)  
Dazee: *****  
Queen: *****  
KBird: *****
```
