

CMSC 201 Section 60

Fall 2024

Sample Final Exam

Answer key

Value: 170 points

Instructions:

This “sample exam” has the same number, type and point values as the questions you will see on the actual final.

On the final, you will have two hours from the start time to complete this exam. You may not use notes, computers, calculators, or other devices. Only pencils or pens.

Section 1: True/False and Multiple Choice. 15 questions; 4 points each. No partial credit is given for questions in this section. Total value: 64 points

1. Which of the following variable names is NOT a valid Python variable name?
 - a. 2CoolforSchool
 - b. 2_Cool_for_School
 - c. Too-Cool-for-School
 - d. None of them is a valid variable name

D, none of the above

2. Which loop would you use if you wanted to get input from the user and stop when the user entered a specific value?
 - a. for I loop
 - b. for each loop
 - c. Sentinel while loop
 - d. Any of the above loops would work

C, sentinel while loop

3. True or False: all input read from files enters as strings, while any immutable type – int, float, Boolean, or string – can be directly read from the keyboard
 - a. True
 - b. False

B, false. Only strings can be read from the keyboard

4. The best way to write a program is:
 - a. Write a little, test a little
 - b. Copy it from some site on the Internet

A, write a little, test a little

5. True or false: the only place in a Python program you CANNOT call a function is on the left side of an assignment statement – the left side of the equals sign.
 - a. True
 - b. False

A, true

6. What is the Python symbol table?
 - a. A list of the values of all variables known to the program at that time
 - b. A data structure that lists the symbols – variables, constants and functions – known to the program at that time, along with their type and the location in memory where the value can be found
 - c. A 2D list in Python
 - d. A 3D list in Python

B

7. Suppose that l is a 2D list. True or False: len(l[1]) tells how many rows are in the list?
 - a. True
 - b. False

B, False. That tells how many columns are in the second row (row 1)

8. Why won't the following recursive program work?

```
def fact(num):
    if num == 1:
        return 1
    else:
        return fact(num)
if __name__ == "__main__":
    n = 6
```

```
result = fact(n)
print (result)
```

- a. There is no base case
- b. There is no recursive case
- c. The recursive case does not result in a simpler problem; that is a problem closer to the base case
- d. What are you talking about? This program works fine.

C. The recursive case results in an infinite loop and a “too many recursions” error

9. Suppose I have a list consisting of the names of the states in the US, like this:

```
states = ['Idaho', 'Utah', 'Hawaii', 'Maine', 'New York', 'South
Carolina', 'Illinois', 'Pennsylvania', 'North Carolina',
'Colorado', 'California', 'Alaska', 'Missouri', 'Kansas',
'Oklahoma', 'Connecticut', 'South Dakota', 'North Dakota',
'Louisiana', 'Nevada', 'Delaware', 'Washington', 'Wisconsin',
'Georgia', 'Nebraska', 'Virginia', 'Wyoming', 'New Hampshire',
'Texas', 'Kentucky', 'West Virginia', 'Rhode Island', 'Maryland',
'Massachusetts', 'Vermont', 'New Mexico', 'Florida', 'Tennessee',
'Iowa', 'Arizona', 'Montana', 'Minnesota', 'Alabama',
'Mississippi', 'Arkansas', 'Oregon', 'New Jersey', 'Ohio',
'Michigan', 'Indiana']
```

How would I create a new list consisting of the first five states in the list, “Idaho” through “New York”?

- a. `new_list = states[:]`
- b. `new_list = states[:6]`
- c. `new_list = states[1:5]`
- d. `new_list = states[:5]`

D. Remember that you start counting at 0.

10. What’s the difference between `readline()` and `readlines()` when you are reading a text file in Python?

- a. `readline()` reads one line of the file, which `readlines()` reads the entire file as a single string
- b. There is no difference; Python does the same for both
- c. `readline()` reads the entire file in as a single string, while `readlines()` reads the file into a list of strings, one list element per line
- d. `readline()` reads the next line of the file as a string, while `readlines()` reads the file into a list of strings, one list element per line.

D,

11. True or False: it is possible to “nest” functions in Python; that is, declare one function to be wholly contained within another function.

- a. True
- b. False

A, true

12. True or False: d, below, is a legal dictionary in Python

- a. True
- b. False

```
d = {  
    "UMBC": "America East",  
    "UMCP": "Big Ten",  
    "Towson": "CAA",  
    "Stony Brook": "CAA",  
    "UMCP": "ACC"  
}
```

B, False. “UMCP” appears twice as a key,
which is not legal

13. Suppose that you have the list

```
l = [1, 2, 3, 4, 5]
```

You want to change l to contain the square of each number. That is, you want l to be [1,4,9,16,25]. True or False: you can use either a “for each” loop or a “for l” loop to accomplish this?

- a. True
- b. False

B, False. for each should not be used if you want to change the values of the list

14. True or false: big Theta of an algorithm exists only if its best-case performance and its worst-case performance are the same.

- a. True
- b. False

A, true

15. Suppose I have a list with the names of the US states in some random order, like:

```
states =
```

```
['Idaho', 'Utah', 'Hawaii', 'Maine', 'New York', 'South  
Carolina', 'Illinois', 'Pennsylvania', 'North Carolina',  
'Colorado', 'California', 'Alaska', 'Missouri', 'Kansas',  
'Oklahoma', 'Connecticut', 'South Dakota', 'North Dakota',  
'Louisiana', 'Nevada', 'Delaware', 'Washington', 'Wisconsin',  
'Georgia', 'Nebraska', 'Virginia', 'Wyoming', 'New Hampshire',  
'Texas', 'Kentucky', 'West Virginia', 'Rhode Island', 'Maryland',  
'Massachusetts', 'Vermont', 'New Mexico', 'Florida', 'Tennessee',  
'Iowa', 'Arizona', 'Montana', 'Minnesota', 'Alabama',  
'Mississippi', 'Arkansas', 'Oregon', 'New Jersey', 'Ohio',  
'Michigan', 'Indiana']
```

Which algorithm would be best to use to determine if “Michigan” is in the list, and if so, what its index is?

- a. Binary search
- b. Linear search
- c. Either binary or linear search will work
- d. Neither binary or linear search will work

B, linear search. Since the list is not sorted, binary search will not work. But linear search will always work.

Section 2: Short answer. Provide a short answer to each question in this section. There are 13 questions in this section, worth 5 points each. Partial credit IS awarded for questions in this section. Total value: 65 points

16. Suppose I have a list `l = [1,2,3,4,5]`. `len(l)` returns the value 5. What happens if I try to print `l[5]`?

You get an “index out of range” error. The last element is always `len(l) - 1`.

18. Suppose I have the following Python statement, designed to get the user to enter a student’s name and test score.

```
data = input("Please enter the student's name and test score. Then hit  
"enter" when done")
```

Write code that will split the user’s input into a string containing the student’s name, and an integer containing the student’s test score.

```
Data_list = data.split()
```

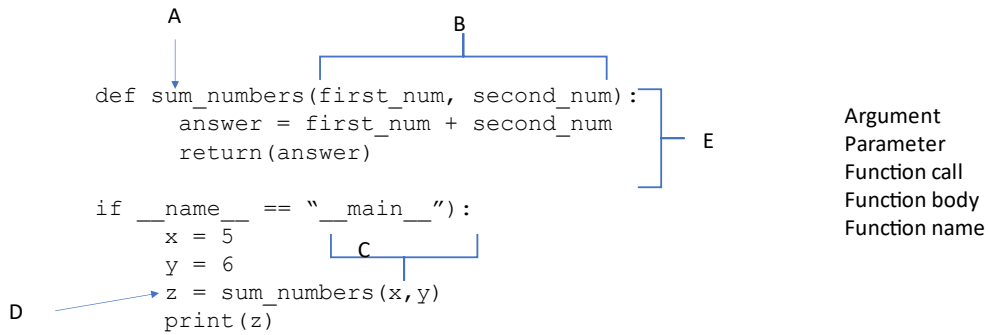
```
Data_list[1] = int(Data_list[1])
```

19. Suppose I have the following Python program. Explain why this would crash with the error “print_result undefined”

```
def solve_problem(n):  
    def print_result( r):  
        number = 5  
        print( number * r)  
  
    answer = n * 5  
    print_result(answer)  
  
def another_problem(m):  
    answer = m *2  
    print_result(answer)  
  
if __name__ == "__main__":  
    y = 2  
    solve_problem(y)  
    another_problem(y)  
    print("We're done")
```

The function `print_result` is only defined within the context of the function `solve_problem`. When `solve_problem` is not running, `print_result` is undefined. When `another_problem` is called, `solve_problem` is not running and thus you get the error messages.

20. Identify each labeled part of the program below:



A=
 B=
 C=
 D=
 E=

A – function name; B – parameter; C – arguments; D- function call; E -function body

21. Why can't you have both a function, length, accessible from the main program and a variable, length, declared in the main program?

Because the symbol table could not distinguish between the two, and Python would potentially get confused.

22. Suppose that `students` is a 2D list with 5 rows and 4 columns. If I try to print `students[1][3]`, which subscript refers to the row number and which subscript refers to the column number?

The first subscript (1) is the row; the second subscript [3] is the column

23. A Python dictionary has keys and values. What types can be legal values in a Python dictionary?

Any type

24. Explain what the “sentinel” is in a sentinel while loop.

The sentinel is the value that causes the loop to stop

25. What’s the difference between a “local variable” and a “global variable” in a Python program?

A global variable is defined in the main program and is accessible anywhere in the program. A local variable is defined in a function and is only accessible from within the function.

26. What happens if you try to open a file in append mode (“a” mode) in Python and the file does not exist?

A new empty file is created for you and given that name

27. Write a recursive function that takes one parameter, a string *s*, and finds if *s* is a palindrome. Your function should return True if *s* is 0 or 1 characters long; it should return False if the first and last letters of *s* are different. Otherwise it should make a recursive call with a substring of *s* that drops the first and last letters from *s*.

You do not need to write the whole program; just the recursive function.

```
def find_palindrome (s):
```

```
    if len(s) <2:
```

```
        return True
```

```
    elif s[0] != s[-1]:
```

```
        return False
```

```
    else:
```



```
return find_palindrome(s[1:-1])
```

28. Explain why the Quicksort algorithm that we studied this semester is almost always faster than bubble sort, even though they have the same worst case behavior.

The “worst case” in Quicksort is incredibly unlikely to happen. Quicksort is far faster in the average case.

29. What does the Big O value of an algorithm such as a sorting algorithm or a searching algorithm tell you?

The approximate performance of the algorithm in the worst case as the size of n goes to infinite.

Section 3: Programming. Write the Python code requested for each of the following problems. Comments are only required when you think what you are doing may not be easy to follow. Three problems; 15 points each. Partial credit IS awarded for questions in this section. Total value: 45 points.

30. Write a Python program that takes the first 100 integers – start at 0; end at 99 – and puts them in a 2D list with 10 rows and 10 columns.

31. Suppose I Have a file, grades.csv, that contains student grade records. The file looks like this:

```
Washington,George,85,82,93,81
```

```
Adams,John,88,77,66,55
```

```
Jefferson,Thomas,92,16,44,0
```

That is, each line contains the information for one student. There are six fields on each line, and those fields are separated by commas. The fields are: last_name, first_name, grade1, grade2, grade3, grade4.

Write a Python program that reads in this file, splits out the fields, adds the four grades to get a total, and prints out to the screen the last name and total grade. Your output should look like:

```
Washington: 341
```

```
Adams:286
```

32. Write a recursive Python function that computes a base raised to the exponent power. The function will have two parameters: base, which is the number to be raised; and exponent, the power to raise the base to. In other words, if we wanted to calculate 2 to the 4th power, we would call this function with 2 as the base and 4 as the exponent. If we wanted to calculate 3 squared, we would call this function with 3 as the base and 2 as the exponent.

If the exponent is 0, your function should just return 1 because any integer to the 0 power is 1. If the exponent is 1, return the base because any number to the first power is that number. If the exponent is greater than 1, return the base times what the recursive call to the function returns.

You may presume that exponent will never be negative.

You do not need to write the entire program; just the recursive function.