CMSC 201 Section 60

Fall 2022

Midterm Exam #2

Date:

November 16, 2002

Value:

100 points

Instructions

The exam will be given in-class. It is open-book; open-note. The exam is subject to UMBC's academic integrity policies. You may NOT solicit help from anyone else during the exam, whether the person is a classmate, another UMBC student, or someone outside the university.

Section 1: Multiple Choice and True/False.

Ten questions, worth 3 points each. Partial credit will NOT be given for questions in this section. There are 30 total points in this section.

1. Which of the following is true about the recursive case in a recursive solution in Python?
    a. There must be one and only one recursive case
    b. The recursive case must result in a simpler problem. That is, it must get you closer to a base case.
    c. A and B are both true
    d. None of the above is true

2. If I have a 2-dimensional list, l, and refer to a cell as l[x][y]. Does x represent the column number or the row number?
    a. The column number
    b. The row number
    c. Neither of these; that's not a valid way to reference a cell in a 2D list in Python
    d. Sorry, I don't have a good fourth option.

3. True or false: lists, like dictionaries, are unordered data structure in Python
   a. True
   b. False

4. Suppose that we have the string

```
s = "antidisestablishmentarianism"
```

What does the following code produce?

```
y = []
for i in range(len(s)):
    if i%2 == 0:
        y.append(s[i])
X = ''.join(y)
print(x)
```

   a. atdssalsmnains
   b. niietbihetraim
   c. Antidisestablishmentarianism
   d. None of the above; the program crashes due to an erro

5. Suppose I have the following dictionary:

```
Schools = {
    "UMBC":"Retrievers",
    "UMCP": "Terrapins",
    "JHU": "Blue Jays"
  }
```

What will be printed if I try to get a value from the dictionary using the statement:

```
print(Schools.get("MSU", -1))
```

   a. Nothing; the program will crash because “MSU” is not a key in the dictionary
   b. “Bears”
   c. None (that is, the special value None of type NoneType)
   d. None of the above is correct

6. Suppose we have the dictionary Schools as defined in problem 5. Now we execute the statement

```
Y = Schools.pop("MSU")
print(Y)
```

What is printed as the result of these statements?

a. None (that is, the special value None of type NoneType)
b. "Bears"
c. Nothing; the program will crash because "MSU" is not a key in the dictionary
d. None of the above is correct

7. True or false: a list variable is a valid key for a Python dictionary
a. True
b. False

8. True or false: you must have exactly one base case in a recursive Python function
a. True
b. False

9. What's wrong with the following Python code?

```
with open("my_results.txt","w") as outfile:
    a = 4
    b = 3.14159
    outfile.write(a, b)
```

a. There's nothing wrong with that code; it will work as presented
b. The file "my_results.txt" doesn't exist, so that will cause an error
c. Only strings can be written to files; not ints or floats. So you can't write a and b without first casting them to strings
d. All of the above are correct; this code is a mess

10. True or false: every problem that can be solved recursively also has an iterative solution, but the reverse is not true
a. True
b. False

Section 2: Short answer.

Eight questions, worth 5 points each. Partial credit WILL be given for answers in this section that are substantially but not completely correct. There are 40 total points in this section.

11. Write Python code to create a dictionary, “Baltimore_colleges” with key/value pairs as follows:

Key: “location” Value: [“Baltimore CIty”, “Baltimore County”]
Key: “schools” value: [“UMBC”, “UMB”, “TSU”, “MSU”, “CSU”, “JHU”]

```
Baltimore_colleges = {
    “location”: [“Baltimore City”, “Baltimore County”],
    “schools”:  [“UMBC”, “UMB”, “TSU”, “MSU”, “CSU”, “JHU”]
        }
```

12. If you open a file in append mode (“a” mode) in Python and the file does not already exist, what happens?

Python creates the file

13. Suppose we have the following dictionary defined:

```
canadian_hockey_teams = {
    "montreal":"canadiens",
    "winnipeg":"jets",
    "edmonton":"oilers",
    "vancouver": "canucks",
    "toronto":"maple leafs",
    "calgary":"flames"
}
```

Let’s suppose the Winnipeg Jets move to Saskatoon and become the Wheatfarmers. Write Python code to remove the Winnipeg entry from this dictionary, and then add a Saskatoon entry with value “Wheatfarmers.”

```
del canadian_hockey_teams[“winnipeg”] # you could use pop
canadian_hockey_teams[“Saskatoon”] = “Wheatfarmers”
```

14. In Python file I/O, what’s the difference between the readline() method and the readlines() method?

readline() reads one line as a string
readlines() reads the rest of the file into a list of strings; one line of the file per list element

15. Suppose you have the following Python program:

```
def check_input(child):
    child *= 4
    return


if __name__ == "__main__":
    d = {
        "name":"Williams",
        "occupation":"plumber",
        "children":2
    }
    check_input(d[“children”])
    print (d)
```

When this is executed, the result is
{'name': 'Williams', 'occupation': 'plumber', 'children': 2}

Explain, using the concept of mutable and immutable types, why the number of children did not change.

16. Explain the difference between the pop() method used for lists and the pop() method used for dictionaries. It is sufficient to describe what each does and point out what’s different.

pop() for lists removes the element by index
pop() for dictionaries removes the specific key/value pair

17. Suppose I have a Python file, “data.txt” that has the following content (note that values on a line are separated by tabs)

```
H     He    Li
Be    B     C
N     O     F
```

I want to read this data into a 2-dimensional list, “elements” with the values on the first line of the file in the first row, the values on the second line of file in the second row, and the values on the third line of the file in the third row. I start with the following code:

```
with open("data.txt", "r") as infile:
      data = infile.read()
```

Write Python code that turns data into the 2D list that I want.

```
# I need to split this string twice: first, I split on "\n"
# then I'm going to split each row on "\t"
# that will give me my 2D list

data_list = data.split("\n")
for i in range(len(data_list)):
      data_list[i] = data_list[i].split("\t")
```

18. Write the Python code to generate a 2-dimensional list, `cubes`, that gives you the cubes of the integers from 0 through 19. Make sure there are 4 rows and 5 columns. The first row should have 0 cubed, 1 cubed, 2 cubed, 3 cubed, and 4 cubed; the second row has 5 cubed,... and so on.

```
cubes = []
for i in range(4):
      new_row = []
      for j in range(5):
            new_row.append((i*5 + j)**3)
      cubes.append(new_row)
```

Section 3: Programming.

Two questions, worth 15 points each. Partial credit WILL be given for answers in this section that are substantially but not completely correct. There are 30 total points in this section.

19. Write a recursive Python function that takes a negative integer and returns the product of all integers between that number and -1. That is, if your function starts with -5, it should ultimately return -5 * -4 *-3 * -2 * -1 = -120. If it starts with -2, it should return -2 * -1 = 2.

You don't need to write the entire program. Just write the recursive function.

```
Def calc_product(num):
      If num == -1:
            Return -1
      Else:
            Return num * calc_product (num + 1)
```

20. I've got a data file, states.txt, that contains the names of the 50 United States in alphabetical order. State names are separated by semi-colons. That is, the file looks like
`Alabama;Alaska;Arizona;Arkansas` and so on.

Write a Python program that reads in this file and separates it into a list, with each state being an element of the list. Then write a loop that goes through this list and prints out the name of each state that begins with the letter "N".

```
With open ("states.txt","r") as infile:
      Data = infile.read()
      State_list = data.split(";")
      For state in state_list:
            If state[0] == "N":
                  print(state)
```