Last Name:
First Name:
NetID:

Computer Science
-
Placement Exam

Instructions

- **Duration**: 45 minutes
- Computers, calculators, phones, textbooks or notebooks are not allowed
- Please use clear indentations in your code, otherwise, grader will consider 'no indentation'.

Question 1 - Predict the Output

The following codes **do not** have any syntax error. **Predict the printed output** obtained by running the following programs.

Program 1	Output 1

```
def fun1():
    print("Let's get going.")
    return 'Great!'
    print("Do you like it?")

def fun2():
    print('Where are we?')
    fun1()

def main():
    fun1()
main()
```

Program 2	Output 2

```
A = {3, 5, 6, 8}
B = {3, 6, 9, 12}
A.add(9)
B.add(9)
print(A.intersection(B))
print(A.union(B))
```

Program 3	Output 3

```
drinks = ['coffee', 'tea', 'orange juice']
try:
    drinks[0] = 3.5
    drinks[1] = 4
    print(drinks[20])
except:
    print('error')
else:
    print('success')
finally:
    print(drinks)
```

Program 4	Output 4

```
L1 = [3,2,1]
L2 = [1,2,3,4]
L1.insert(0,4)
D1 = { 'ICP': 'A' , \
       'Operating Systems': 'A',\
       'Data Structures': 'B' }
D2 = { 'Data Structures': 'B', \
       'ICP': 'A' ,\
       'Operating Systems': 'A', \
       'Algorithms': 'C' }
del D2['Algorithms']
S1 = {1,1,2,2,3,3,4}
S1.add(4)
S2 = {4,3,2,1}
print(L1==L2, D1==D2, S1==S2)
```

Program 5	Output 5

```
reg_guest = ['Judy2002', 'Sam2002', \
             'Tim2010', 'Wang2009']
for g in reg_guest:
    if '2002' in g:
        reg_guest = reg_guest.remove(g)
print(reg_guest)
```

Question 2 - Boolean Conditions

Evaluate the following Boolean expressions:

- `5 ** 2 >= 20` ⟶ _____
- `'n' == 'na'` ⟶ _____
- `not True and False` ⟶ _____
- `False and True or True` ⟶ _____
- `not True or not False and True` ⟶ _____

Question 3 - "Big" Countries

In a file named `world.txt`, there is a **list of countries with some information**.
The first few lines are shown below for you to **check the format**.

`world.txt` (only the first 6 lines are shown)

```
country,continent,area(sq km),population,gdp
Afghanistan,Asia,652230,25500100,20343000
Albania,Europe,28748,2831741,12960000
Algeria,Africa,2381741,37100000,188681000
Andorra,Europe,468,78115,3712000
Angola,Africa,1246700,20609294,100990000
```

`str` `country,continent,area,population,gdp\nAfghanistan,Asia,652230,25500100...`

3.1 Task 1 - Country Info

Write a function named `country_info` that:

- takes a filename (type: `str`) as a **parameter** (for example `'world.txt'`)
- and **returns a dictionary**.
- The dictionary format is:
  - `key` → **country name** (type: `str`)
  - `value` → **a list** (type: `list`) containing 5 items (in this order):
    1. continent (type: `str`)
    2. area in km^2 (type: `int`)
    3. population (type: `int`)
    4. GDP (type: `int`)
    5. a boolean value (type: `bool`) representing if the country is defined as "big" or not

Definition of "big" country:
A country is **"big"** if it has an area of ***at least 3 million km^2*** **OR** a population of ***at least 25 millions***.

Example of **returned dictionary** if one reads the **first 6 lines of the file only**:

```
{'Afghanistan': ['Asia', 652230, 25500100, 20343000, True],
'Albania': ['Europe', 28748, 2831741, 12960000, False],
'Algeria': ['Africa', 2381741, 37100000, 188681000, True],
'Andorra': ['Europe', 468, 78115, 3712000, False],
'Angola': ['Africa', 1246700, 20609294, 100990000, False]}
```

Write the `country_info` function here

3.2 Task 2 - "Big" Countries per continent

Write a function named `big_countries` that:

- takes a dictionary (type: `dict`) as a **parameter**
  - the format is similar to the one obtained in Task 1
- and **returns another dictionary**.
- The returned dictionary format is:
  - `key` → **continent** (type: `str`)
  - `value` → **a list of strings** (type: `list`) containing the **countries** (type: `str`) defined as **"big"** in that continent

Example of **returned dictionary** if one uses the **example dictionary from previous task**:

```
{'Asia': ['Afghanistan'], 'Africa': ['Algeria']}
```

Write the `big_countries` function here