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.

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Output 1</th>
</tr>
</thead>
</table>
| ```python
1 def fun1():
2     print("Let's get going.")
3     return 'Great!'
4     print("Do you like it?"
5
6 def fun2():
7     print('Where are we?')
8     fun1()
9
10 def main():
11     fun1()
12     main()
``` |           |

<table>
<thead>
<tr>
<th>Program 2</th>
<th>Output 2</th>
</tr>
</thead>
</table>
| ```python
1 A = {3, 5, 6, 8}
2 B = {3, 6, 9, 12}
3 A.add(9)
4 B.add(9)
5 print(A.intersection(B))
6 print(A.union(B))
``` |           |

<table>
<thead>
<tr>
<th>Program 3</th>
<th>Output 3</th>
</tr>
</thead>
</table>
| ```python
1 drinks = ['coffee', 'tea', 'orange juice']
2 try:
3     drinks[0] = 3.5
4     drinks[1] = 4
5     print(drinks[20])
6 except:
7     print('error')
8 else:
9     print('success')
10 finally:
11     print(drinks)
``` |           |
Question 2 - Boolean Conditions

Evaluate the following Boolean expressions:

- $5 \times 2 \geq 20 \rightarrow \underline{\quad}$
- 'n' == 'na' \rightarrow \underline{\quad}
- not True and False \rightarrow \underline{\quad}
- False and True or True \rightarrow \underline{\quad}
- not True or not False and True \rightarrow \underline{\quad}
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.

<table>
<thead>
<tr>
<th>world.txt (only the first 6 lines are shown)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  country,continent,area(sq km),population,gdp</td>
</tr>
<tr>
<td>2  Afghanistan,Asia,652230,25500100,20343000</td>
</tr>
<tr>
<td>3  Albania,Europe,28748,2831741,12960000</td>
</tr>
<tr>
<td>4  Algeria,Africa,2381741,37100000,188681000</td>
</tr>
<tr>
<td>5  Andorra,Europe,468,78115,3712000</td>
</tr>
<tr>
<td>6  Angola,Africa,1246700,20609294,100990000</td>
</tr>
</tbody>
</table>

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 km² (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 km² OR a population of at least 25 millions.

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

```python
{'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