CS 210: Data Management for Data Science

Sample Midterm 1

Fall 2023

Name: ____ NetID: ____________________________

This is a closed book, closed notes exam, only 5 pages of HAND WRITTEN NOTES allowed.

No electronic devices are permitted.

Name: ____________________________

1. (10 points) The Rutgers libraries need your help organizing their books. They have a list of texts that they would like to be organized by the number of words that exist in the book. A word is defined by a string that is surrounded by delimiters (white space, newline, etc.) on each side. Given a list of files to read in as an argument, return the book list in order from books with the greatest to least words.

   ```
   def organize_books(books: list) -> list:
   ```

2. (10 points) Write a Python function that takes a string as input and returns the longest substring without repeating characters. For example, if the input string is **'abcabcbb'**, the function should return **'abc'** because it is the longest substring without repeating characters.

Name: ____________________________

3. (10 points) Write a password generator in Python. The passwords should be random, generating a new password every time the user asks for a new password. You are given three strings, one contains special characters, one has all the lowercase letters, and one has all the uppercase letters. The password needs to contain the following:

   - Must be 8-20 characters long
   - At least 1 lowercase alphabet from given string of `lowecase_letters`
   - At least 1 uppercase alphabet from given string of `uppercase_letters`
   - At least 1 number [0-9]
   - At least 1 special character from the given string of special characters

```
def generate_pass() -> str:
    special_characters = "!@#$%^&*"
    lowercase_letters = "abcde... z"
    uppercase_letters = "ABCDE... Z"
```

4. (10 points) Given a list of integers **nums** and an integer **target**, write a Python function to find a contiguous subarray (a subset of the list) that, when summed, equals the **target**. Return the starting and ending indices of the subarray. For example, if **nums** = [2, 7, 11, 15] and **target** = 9, the function should return [0, 1] because nums[0] + nums[1] = 2 + 7 = 9.

5. (10 points) Given a dictionary where the key is the class name and the value is the number of students in the class, answer the following questions. An example of the dictionary is shown below:

```
{
    "CS210": 540,
    "EE279": 250,
    "PHIL101": 440 ...
}
```

(a) Write a function that takes the above dictionary as an argument and returns a list of the top 10 largest classes at Rutgers, these classes must be in order from largest to smallest.

```
def find_top_ten(classes: dict) -> list:
```

(b) The CS department wants to know which upper level classes are unpopular, an upper level class is defined as a class with a level ≥ 300, (the level is the last three digits of the key e.x. the 450 in CS450). Given the dictionary above, find the ten CS classes that are higher than or equal to 300 level which have the lowest enrollment, then return these classes as a list.

```
def find_unpopular(classes: dict) -> list:
```

6. (10 points)You have a CSV file named **employeeData.csv** with the following columns: EmployeeID, FirstName, LastName, Department, Salary, and HireDate. Each row represents an employee's information in a company.

   (a) Write Python code to read the **employeeData.csv** file into a Pandas DataFrame named "employee_df."

   (b) Calculate and print the following statistics for the Salary column:
   Mean (average) salary, Median salary and Standard deviation of salaries.

   (c) Find and print the following information:
   1. The employee with the highest salary, including their full name and salary.
   2. The department with the most employees.

   (d) Create a new column named YearsWorked in the DataFrame, which represents the number of years each employee has been with the company. Assume the current year is 2023, and calculate the YearsWorked based on the HireDate column. Then, print the updated DataFrame with the new column included.

7. (10 points) It's bonus day at your sporting goods employer's workspace, they have two tables of employee data and want you to return the following results so that they can assign bonuses correctly. The **Employee Sales** table contains the amount of sales that each employee made in the past year, as well as their costs (gas, food, etc). The **Employee Info** table contains info about the employee such as their department, email, and phone number.

Employee ID	Sales	Costs
139	1367	198
170	10967	507

Table 1: Employee Sales

Employee ID	Name	Department Name	Email	Phone
139	Fishing	Bobby Jindhal	b.jindhal@rocketmail.com	898-078-9475
170	Skiing	Noah Baumbach	nbthegoat@gmail.com	947-958-0973

Table 2: Employee Info

(a) Write pandas code to join the two DataFrames together, so that each row in your new DataFrame will contain both the sales of the employee as well as their personal information.

```
import pandas as pd

employee_sales = pd.df(... some data)
employee_info = pd.df(... some more data)
```

(b) HR wants to better understand where the company's employees are located to do this they want to create a new column in the DataFrame that contains the area code from each employee's phone number as an integer. If the employee's phone number is null, the area code also needs to be a null or NaN. You can name this new column anything you would like.

(c) Management would like to throw the best performing department an HR party. Performance here is calculated as the sales of all the people in the department minus the costs of each person. Store the name of the department in the `winner` variable.

```
winner = None # store the name of the winning department
```

8. (10 points) You have a Pandas DataFrame named product_df with the following columns: ProductID, ProductName, Category, Price, and InStock. The DataFrame contains information about various products.

   (a) Create a new DataFrame named affordable_products_df containing only the products with a price less than $50 and are currently in stock.

   (b) Calculate and create a new DataFrame named category_summary_df with two columns: Category and Total Products. The Category column should contain unique product categories, and the Total Products column should contain the count of products in each category.

9. (10 points) You're given a dataframe called `daily_temp_df` with two columns, `Date` which is a string with the following format mm/dd/yyyy, and `Average Temp` which contains the average temperature for that day. The latest temperature you have is for date 09/30/2023 and the earliest is for 01/01/2000 The table looks something like this:

Date	Average Temp
09/30/2023	78.9
...	...
01/01/2000	35.6

(a) We want to first find the average weekly temperature between the first date and the latest date in the dataframe. Assume that The first date is a Sunday and the latest is a Saturday (there are 7 days in each week), write pandas code to calculate the rolling average such that you find the average weekly temp. The resulting dataframe should look like this:

Week	Average Weekly Temp
1	80.4
...	...
1196	32.6

(b) You're curious to find if your data contains evidence for climate change, calculate the average temperature for the month of august in each year, then check if the average temperature is increasing each year. If in each consecutive year the average temperature in August is higher than the year before, then store `True` in the `result`, otherwise store `False`. **Hint:** To solve this question I would split the date value into new columns containing the day, month, and year, then I would filter out values I don't need and group accordingly.

```
result = None # store the result of your climate change calculations
```

Name: ____________________________

10. (10 points) You have a dataset containing the monthly sales figures (in dollars) for a retail store over a year. Your task is to perform various operations using NumPy.

```
sales_data = [12000, 13500, 11000, 14500, 15000, 13200, 11800, 15200, 16200,
17500, 18000, 14000]
```

(a) Create a NumPy array named sales_data from the dataset above.

(b) Calculate and print the monthly growth rate for sales as a percentage. The growth rate for a particular month is calculated as:

$$\text{Growth Rate} = \left(\frac{\text{Sales in Current Month} - \text{Sales in Previous Month}}{\text{Sales in Previous Month}}\right) \times 100$$

(c) Find and print the month with the highest sales and the corresponding sales amount.