```python
# import package
import pandas as pd
# creating and loading a dataframe named df_users by reading user_table csv file from the disk
# csv file is in the same folder as python file
df_users = pd.read_csv(r'user_table.csv')
# setting the console display config
pd.set_option('display.width', 320)
pd.set_option('display.max_columns', 11)
# 1.1
print("\nWhat is the number of unique name combinations?")
# finding unique names in dataframe df_users by using pandas drop_duplicate function
# creating unique name dataframe with duplicate rows removed
# by only considering columns Surname and Name
# and keeping only the first occurrence of duplicate records
unique_name = df_users.drop_duplicates(subset=['Surname', 'Name'], keep='first')
print(unique_name)
# counting and printing the unique name combination value by len function
print("\n- The number of unique name combination is " + str(unique_name.__len__()))
# 1.2
print("\nWho is the oldest user, who is the youngest?")
# extracting oldest user by age in the dataframe df_users by using max function
# extracting youngest user by age in the dataframe df_users by using min function
max_age = df_users[['Surname', 'Name']][df_users.Age == df_users.Age.max()]
min_age = df_users[['Surname', 'Name']][df_users.Age == df_users.Age.min()]
# print(max_age.shape) is returning 1 record
# print(min_age.shape) is returning 2914 record
# to distinguish only 1 youngest user, considering the user who has minimum age and who subscribed last
# sorting the dataframe by using sort_value function on Age and Subscription_Date column
# keeping the dataframe in descending order
df_sorted = df_users.sort_values(by=['Age', 'Subscription_Date'], ascending=False)
print(df_sorted)
# printing the oldest user in Name Surname format with maximum Age value
```

```
# retrieving corresponding values by using iloc function (row and columns by index positions)
print("\n- The oldest user is " + str(df_sorted.iloc[0, 3]) + " " + str(df_sorted.iloc[0, 2]) + " with Age " +
      str(df_sorted.iloc[0, 4]))
# printing the youngest user in Name Surname format with minimum Age value
# retrieving corresponding values by using iloc function (row and columns by index positions)
print("- The youngest user is " + str(df_sorted.iloc[-1, 3]) + " " + str(df_sorted.iloc[-1, 2]) + " with Age " +
      str(df_sorted.iloc[-1, 4]) + "\n")
# 2.1
# creating and loading a dataframe named df_posts by reading postings_table csv file from the disk
# csv file is in the same folder as python file
df_posts = pd.read_csv(r'postings_table.csv')
# renaming the column name ID to UserID in df_users dataframe
df_users.rename(columns={'ID': 'UserID'}, inplace=True)
# merging both the dataframes (df_users and df_posts) on UserID column by merge function
df_part2 = pd.merge(df_users, df_posts, on='UserID')
# dropping the unnamed index columns by drop function in the newly merged dataframe df_part2
df_part2.drop(['Unnamed: 0_x', 'Unnamed: 0_y'], axis='columns', inplace=True)
# creating a new dataframe df_posting and grouping the data by UserID, Surname and Name
# counting and sorting the number of post by each user on grouped data
# through count and sort value function on PostID column
# keeping the dataframe in descending order on PostID and resetting the dataframe index
df_posting = df_part2.groupby(['UserID', 'Surname', 'Name']).count().sort_values('PostID', ascending=False)['PostID'].reset_index()
print("User Postings")
# dataframe df_posting is displaying the count of each users posting
print(df_posting)
# only 1 user with maximum number of posting
# but there are multiple users with least number of postings
print("\nWho is the user with most postings?")
# printing the user with most postings in Name Surname format with total number of posts
# retrieving corresponding values by using iloc function (row and columns by index positions)
```

```
63 print("- The user with most postings is " + str(df_posting.iloc[0, 2]) + " " + str(df_posting.iloc[0
   , 1]) +
64       ",No. of postings: " + str(df_posting.iloc[0, -1]))
65 # 2.2
66 print("\nWho has the least amount of postings?")
67 # 42 users are with the least amount of posting which is 1 we can print all of the users name whose
   posting count = 1
68 print("- The users with least postings " + "(No. of postings: " + str(df_posting.iloc[-1, -1]) + ")"
    + " are:")
69 # retrieving the value of least_postings from PostID column in df_posting dataframe by using iloc
   function
70 # accessing the row and column by index positions
71 # using for loop to retrieve and iterating over each row by using iterrow function
72 # printing all the users in Name Surname format with least posting count
73 least_postings = df_posting[df_posting['PostID'] == df_posting.iloc[-1, -1]]
74 for index, row in least_postings.iterrows():
75     print(row['Name'], row['Surname'])
76 # 2.3
77 # creating a new column Wordcount in df_part2 dataframe
78 # counting the words for each post in Content column by applying lambda function
79 # to retrieve length of the strings and splitting by space
80 df_part2['Wordcount'] = df_part2['Content'].apply(lambda x: len(str(x).split(' ')))
81 # creating a new dataframe df_totalwords and using groupby function on UserID, Surname and Name
   columns
82 # finding the total Wordcount for each grouped UserID by using sum function on Wordcount column
83 # keeping the dataframe in descending order on Wordcount and resetting the dataframe index
84 df_totalwords = df_part2.groupby(['UserID', 'Surname', 'Name']).sum().sort_values('Wordcount',
   ascending=False)['Wordcount'].reset_index()
85 print("\nUser written words")
86 print(df_totalwords)
87 # only 1 user is written the most words and multiple users are with the least written words
88 print("\nWhich user has written most words?")
89 # printing the user with most written words in Name Surname format with number of total written words
```

```
90 # retrieving corresponding values by using iloc function (row and columns by index positions)
91 print("- " + str(df_totalwords.iloc[0, 2]) + " " + str(df_totalwords.iloc[0, 1]) +
92       " is the user with most written words with wordcount " + str(df_totalwords.iloc[0, -1]))
93 # 2.4
94 print("\nWhich one has written the least?")
95 # 8 users are with the least written words which is 21 we can print all of the corresponding users
   name
96 print("- The users with least written words " + "(wordcount: " + str(df_totalwords.iloc[-1, -1]) +
   ")" + " are:")
97 # retrieving the value of least written words from Wordcount column in df_totalwords dataframe by
   using iloc function
98 # accessing the row and columns by index positions
99 # using for loop to retrieve and iterating over each row by using iterrow function
100 # printing all the users in Name Surname format with least written words
101 least_written = df_totalwords[df_totalwords['Wordcount'] == df_totalwords.iloc[-1, -1]]
102 for index, row in least_written.iterrows():
103     print(row['Name'], row['Surname'])
104 print("\nTask Done!")
105
```