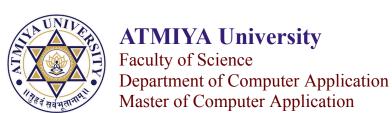


Enrollment No:=230823007 | Name:-Jainish Barbhaya

## **Practice Set 6**

1.) Create a file with file name sample.txt, accept some data from the user and store it in the file. filename = "sample.txt" data = input("Enter data to store in the file: ") with open(filename, 'w') as file: file.write(data) output:= Enter data to store in the file: hiiiiiiiii helo hoe are you. 2.) Display the data stored in the sample.txt file (created in question 1). with open (filename, 'r') as file: file data = file.read() print("Data stored in sample.txt:", file data) output:= Data stored in sample.txt: hiiiiiiiiii helo hoe are you.

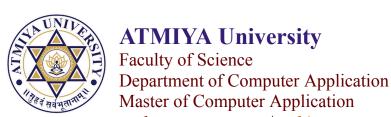


Enrollment No:=230823007 | Name:-Jainish Barbhaya

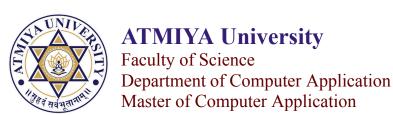
3.) Accept some data from the user and append it into the file sample.txt (created in question 1), also the data in the file. additional data = input("Enter additional data to append: ") with open(filename, 'a') as file: print("Data appended in sample.txt:", additional data) output:= Enter additional data to append: hello good morning mojma. Data appended in sample.txt: hello good morning mojma. 4.) Accept the file name from the user, check the availability of the file: i). If the file exists display the data on the screen, ii). If the file is not available, display the appropriate message. user file = input("Enter the file name to check: ") try: with open(user file, 'r') as file: file\_data = file.read() print("Data stored in", user file + ":", file data) except FileNotFoundError: print("File not found!") output:= Data stored in sample.txt: hiiiiiiiiii helo hoe are you. hiiiiiiiii helo hoe are you.



```
hi jbnm
lpmnjklo
5.) Accept the file name from the user, check the availability
of the file:
a. If the file exists, display: i). No. of characters, ii). No.
of words and iii). No. of lines
b. If the file does exist, than display the appropriate message.
user file = input("Enter the file name to check: ")
try:
    with open(user file, 'r') as file:
        char count = len(file.read())
        file.seek(0)
        word count = len(file.read().split())
        file.seek(0)
        line count = len(file.readlines())
        print("File characteristics:")
        print("Number of characters:", char count)
        print("Number of words:", word count)
        print("Number of lines:", line_count)
except FileNotFoundError:
   print("File not found!")
output:=
Enter the file name to check: sample.txt
File characteristics:
Number of characters: 79
```



```
Number of words: 13
Number of lines: 4
6.) Create and open the binary file with 'with' option. Store
names of all the subjects you study in
semester 2. Ask user to enter the subject number they wanted to
see and display that subject
name.
subjects = ["Java", "Python", "Laravel", "AWS", "Language and
Numeric Method"] # Example subjects
with open('semester2 subjects.bin', 'wb') as file:
    for subject in subjects:
        file.write(subject.encode() + b'\n')
subject number = int(input("Enter the subject number you want to
see: "))
with open('semester2 subjects.bin', 'rb') as file:
    subjects = file.readlines()
    if 1 <= subject number <= len(subjects):</pre>
        print(f"Subject {subject number}:
{subjects[subject number - 1].decode().strip()}")
    else:
        print("Invalid subject number.")
output:=
Enter the subject number you want to see: 5
Subject 5: Language and Numeric Method
```



zipf.write('img2')

Code:-23PGCA207 | Subject-Name:- Programming With Python

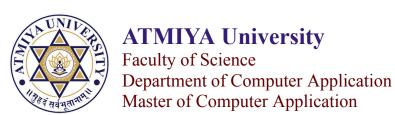
Enrollment No:=230823007 | Name:-Jainish Barbhaya

7.) Create a file named 'img1', store an image into it. Open another file named 'img2', copy the same image as in the file 'img1'. Also store both files into the zip file named 'imp img'. import shutil import zipfile # Step 1: Store an image into 'img1' # Assuming 'image.jpg' is the image file you want to store shutil.copy('image.png', 'img1') # Step 2: Open another file 'img2' and copy the same image as in 'img1' shutil.copy('img1', 'img2') #Step 3: Store both 'img1' and 'img2' files into the zip file 'imp img' with zipfile.ZipFile('imp img.zip', 'w') as zipf: zipf.write('img1')

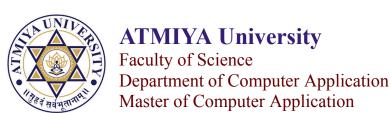
- 8.)Create a file with 'with' option, name it as 'marks.dat'.
- i). Accept subject name and marks from the user, store the data in the file.
- ii). Give three options to the user: a). To view whole file, b).Accept and edit the marks of a subject user want to change.iii). Exit



```
with open('marks.dat', 'w') as file:
    while True:
        choice = input("Enter subject name and marks you want to
enter or enter 'exit' to quit: ")
        if choice.lower() == 'menu':
            print("Menu:\n1. View whole file\n2. Edit marks of a
subject\n3. Exit")
        elif choice.lower() == 'exit':
            break
        else:
            file.write(choice + '\n')
# Display menu options
print("Menu:\n1. View whole file\n2. Edit marks of a subject\n3.
Exit")
# Handle user choices
while True:
    option = input("Enter your choice (1, 2, or 3): ")
    if option == '1':
        with open('marks.dat', 'r') as file:
            print(file.read())
    elif option == '2':
        subject to edit = input("Enter the subject name you want
to edit: ")
        new marks = input("Enter the new marks: ")
        with open('marks.dat', 'r') as file:
            lines = file.readlines()
        with open('marks.dat', 'w') as file:
```

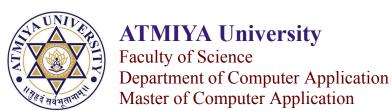


```
for line in lines:
                if subject to edit in line:
                    file.write(subject_to_edit + ',' + new_marks
+ '\n')
                else:
                    file.write(line)
    elif option == '3':
        print("Exiting...")
        break
    else:
        print("Invalid choice. Please enter 1, 2, or 3.")
output:=
Enter subject name and marks you want to enter or enter 'exit'
to quit: MATH, 90
Enter subject name and marks you want to enter or enter 'exit'
to quit: PYTHON, 80
Enter subject name and marks you want to enter or enter 'exit'
to quit: AWS,70
Enter subject name and marks you want to enter or enter 'exit'
to quit: LARAVEL, 60
Enter subject name and marks you want to enter or enter 'exit'
to quit: JAVA,50
Enter subject name and marks you want to enter or enter 'exit'
to quit: J2EE,40
Enter subject name and marks you want to enter or enter 'exit'
to quit: C++,30
Enter subject name and marks you want to enter or enter 'exit'
to quit: exit
```

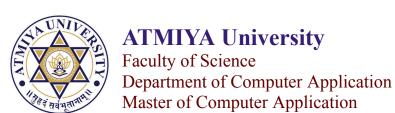


```
Menu:
```

```
1. View whole file
2. Edit marks of a subject
3. Exit
Enter your choice (1, 2, or 3): 1
MATH, 90
PYTHON,80
AWS, 70
LARAVEL, 60
JAVA,50
J2EE, 40
C++,30
Enter your choice (1, 2, or 3): 2
Enter the subject name you want to edit: SOFTWARE
Enter the new marks: 78
Enter your choice (1, 2, or 3): 3
Exiting...
9. Create a regular expression that:
a). Identifies and display the string starting with 's' and
having 4 characters.
import re
text = "Sample text with some words starting with s and having
four characters such as sand, silt."
matches a = re.findall(r'\bs\w{3}\b', text)
print("Strings starting with 's' and having four characters:",
matches a)
```



```
output:=
Strings starting with 's' and having four characters: ['some',
'such', 'sand', 'silt']
b). Splits the string where some special characters are found.
import re
text = "This!is,a@sample;text#with:special/characters"
splitted = re.split(r'[!@#;,/:]', text)
print("String split where special characters are found:",
splitted)
output:=
String split where special characters are found: ['This', 'is',
'a', 'sample', 'text', 'with', 'special', 'characters']
c). Display the word starting with number.
import re
text = "Sample text with 123 numbers and some words."
matches c = re.findall(r'\b\d\w*\b', text)
print("Words starting with a number:", matches c)
output:=
Words starting with a number: ['123']
```



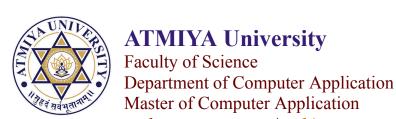
Enrollment No:=230823007 | Name:-Jainish Barbhaya

d). Display the word having 3 or 4 or 5 characters.

```
import re
text = "Sample text with some short and long words."
matches d = re.findall(r'\b\w{3,5}\b', text)
print("Words having 3, 4, or 5 characters:", matches d)
output:=
Words having 3, 4, or 5 characters: ['text', 'with', 'some',
'short', 'and', 'long', 'words']
e). Display only the dates from the string.
import re
text = "Today is 2024-04-18 and tomorrow will be 2024-04-19."
dates = re.findall(r'\b\d{4}-\d{2}-\d{2}\b', text)
print("Dates from the string:", dates)
output:=
Dates from the string: ['2024-04-18', '2024-04-19']
f). Create a string with name of the person and his Aadhar
number, display only Aadhar
number.
import re
text = "Name: John Doe, Aadhar: 1234 5678 9012"
```



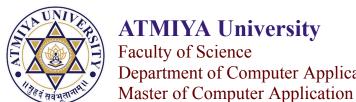
```
aadhar\_number = re.search(r'\b\d{4}\s\d{4}\s\d{4}\b',
text) .group()
print("Aadhar number:", aadhar_number)
output:=
Aadhar number: 1234 5678 9012
g). Display all the words that starts with 'at' or 'ap'.
import re
text = "Sample text with words like apple, at, bat, and cat."
matches g = re.findall(r'\b[ap]t\w*\b', text)
print("Words starting with 'at' or 'ap':", matches g)
output:=
Words starting with 'at' or 'ap': ['at']
h). Check if the string starts with 'at' than display
appropriate message and otherwise.
import re
text = "Sample text starting with at."
if re.match(r'^at', text):
   print("String starts with 'at'.")
else:
   print("String does not start with 'at'.")
```



```
output:=
String does not start with 'at'.
10. Do as directed:
a). Name the package used to deal with data frame.
import pandas as pd
b). Name the package used to deal with data .xlsx file.
import pandas as pd
c). Name the function used to read the .csv file.
import pandas as pd
# Example usage:
data = pd.read csv('data.csv')
d). Name the function used to read the .xlsx file.
import pandas as pd
Example usage:
data = pd.read excel('data.xlsx')
e). Name the function used to read the tuple.
import pandas as pd
# Example usage:
tuple data = [('John', 25), ('Alice', 30), ('Bob', 28)]
data = pd.DataFrame(tuple_data, columns=['Name', 'Age'])
```



```
11. Create a dictionary which stores (at least 10 records) empid,
name, city, salary and perform
following operations:
a). Display first three records
b). Display last five records
c). Display only Name and City
d). Display employee who belongs to Mumbai
e). Display employee name who belongs to Mumbai
f). Display employee whose salary is more than 25000
employee data = [
    {'empid': 1, 'name': 'John', 'city': 'Mumbai', 'salary':
30000},
    {'empid': 2, 'name': 'Alice', 'city': 'Delhi', 'salary':
28000},
    {'empid': 3, 'name': 'Bob', 'city': 'Chennai', 'salary':
32000},
    {'empid': 4, 'name': 'Emily', 'city': 'Mumbai', 'salary':
26000],
    {'empid': 5, 'name': 'David', 'city': 'Kolkata', 'salary':
35000},
    {'empid': 6, 'name': 'Sophia', 'city': 'Mumbai', 'salary':
27000},
    { 'empid': 7, 'name': 'Emma', 'city': 'Pune', 'salary':
29000},
    {'empid': 8, 'name': 'James', 'city': 'Bangalore', 'salary':
31000},
    {'empid': 9, 'name': 'Liam', 'city': 'Mumbai', 'salary':
30000},
```



## **ATMIYA University**

Faculty of Science Department of Computer Application

Code:-23PGCA207 | Subject-Name:- Programming With Python

```
{'empid': 10, 'name': 'Olivia', 'city': 'Hyderabad',
'salary': 26000}
1
# Display first three records
print("a). First three records:")
for emp in employee data[:3]:
   print(emp)
# Display last five records
print("\nb). Last five records:")
for emp in employee data[-5:]:
   print(emp)
# Display only Name and City
print("\nc). Displaying only Name and City:")
for emp in employee data:
   print(f"Name: {emp['name']}, City: {emp['city']}")
# Display employee who belongs to Mumbai
print("\nd). Employees who belong to Mumbai:")
for emp in employee data:
    if emp['city'] == 'Mumbai':
        print(emp)
# Display employee name who belongs to Mumbai
print("\ne). Employee names who belong to Mumbai:")
for emp in employee data:
    if emp['city'] == 'Mumbai':
        print(emp['name'])
```



```
# Display employee whose salary is more than 25000
print("\nf). Employees whose salary is more than 25000:")
for emp in employee data:
    if emp['salary'] > 25000:
        print(emp)
output:=
a). First three records:
{'empid': 1, 'name': 'John', 'city': 'Mumbai', 'salary': 30000}
{'empid': 2, 'name': 'Alice', 'city': 'Delhi', 'salary': 28000}
{'empid': 3, 'name': 'Bob', 'city': 'Chennai', 'salary': 32000}
b). Last five records:
{'empid': 6, 'name': 'Sophia', 'city': 'Mumbai', 'salary':
27000}
{'empid': 7, 'name': 'Emma', 'city': 'Pune', 'salary': 29000}
{'empid': 8, 'name': 'James', 'city': 'Bangalore', 'salary':
31000}
{'empid': 9, 'name': 'Liam', 'city': 'Mumbai', 'salary': 30000}
{'empid': 10, 'name': 'Olivia', 'city': 'Hyderabad', 'salary':
26000}
c). Displaying only Name and City:
Name: John, City: Mumbai
Name: Alice, City: Delhi
Name: Bob, City: Chennai
Name: Emily, City: Mumbai
Name: David, City: Kolkata
Name: Sophia, City: Mumbai
```



```
Name: Emma, City: Pune
Name: James, City: Bangalore
Name: Liam, City: Mumbai
Name: Olivia, City: Hyderabad
d). Employees who belong to Mumbai:
{'empid': 1, 'name': 'John', 'city': 'Mumbai', 'salary': 30000}
{'empid': 4, 'name': 'Emily', 'city': 'Mumbai', 'salary': 26000}
{'empid': 6, 'name': 'Sophia', 'city': 'Mumbai', 'salary':
27000}
{'empid': 9, 'name': 'Liam', 'city': 'Mumbai', 'salary': 30000}
e). Employee names who belong to Mumbai:
John
Emily
Sophia
Liam
f). Employees whose salary is more than 25000:
{'empid': 1, 'name': 'John', 'city': 'Mumbai', 'salary': 30000}
{'empid': 2, 'name': 'Alice', 'city': 'Delhi', 'salary': 28000}
{'empid': 3, 'name': 'Bob', 'city': 'Chennai', 'salary': 32000}
{'empid': 4, 'name': 'Emily', 'city': 'Mumbai', 'salary': 26000}
{'empid': 5, 'name': 'David', 'city': 'Kolkata', 'salary':
35000}
{'empid': 6, 'name': 'Sophia', 'city': 'Mumbai', 'salary':
27000}
{'empid': 7, 'name': 'Emma', 'city': 'Pune', 'salary': 29000}
{'empid': 8, 'name': 'James', 'city': 'Bangalore', 'salary':
31000}
```



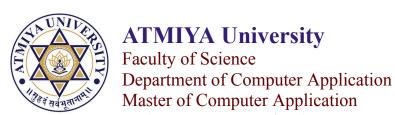
```
{'empid': 9, 'name': 'Liam', 'city': 'Mumbai', 'salary': 30000}
{'empid': 10, 'name': 'Olivia', 'city': 'Hyderabad', 'salary':
26000}
12. Create an xlsx file store marks of five subjects, plot the
data on the bar graph.
import pandas as pd
import matplotlib.pyplot as plt
marks data = {
    'Student1': [80, 75, 85, 90, 88],
    'Student2': [70, 65, 75, 80, 78],
    'Student3': [85, 80, 88, 92, 90],
    'Student4': [78, 82, 79, 85, 80],
    'Student5': [92, 88, 90, 85, 95]
}
df = pd.DataFrame(marks data, index=['Maths', 'Physics',
'Chemistry', 'Biology', 'English'])
df = df.T
df.to excel('marks_data.xlsx')
df.plot(kind='bar', figsize=(10, 6))
plt.title('Student Marks in Five Subjects')
plt.xlabel('Students')
plt.ylabel('Marks')
plt.xticks(rotation=45)
```



```
plt.legend(title='Subjects')
plt.grid(axis='y')
plt.tight layout()
plt.show()
13. Take five income source of the Government and display it on
the pie chart.
import matplotlib.pyplot as plt
income sources = ['Taxes', 'Fees and Charges', 'Grants',
'Investment Income', 'Other']
income values = [40, 20, 15, 10, 15]
explode = (0.1, 0, 0, 0, 0)
plt.figure(figsize=(8, 6))
plt.pie(income values, labels=income sources, explode=explode,
autopct='%1.1f%%', startangle=140)
plt.title('Government Income Sources')
plt.axis('equal')
plt.show()
14. Draw the line chart representing BSE (Bombay Stock Exchange)
index in last 10 years.
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
```



```
start date = datetime.now() - timedelta(days=10*365)
end date = datetime.now()
date range = pd.date range(start=start date, end=end date,
freq='D')
bse index = pd.Series(index=date range,
data=np.random.randint(20000, 50000, len(date_range)))
plt.figure(figsize=(10, 6))
plt.plot(bse index.index, bse index.values, color='blue',
marker='o', linestyle='-')
plt.title('BSE Index Over the Last 10 Years')
plt.xlabel('Date')
plt.ylabel('BSE Index')
plt.xticks(rotation=45)
plt.tight layout()
plt.grid(True)
plt.show()
15. Plot the grouped bar graph using the appropriate data.
import matplotlib.pyplot as plt
import numpy as np
categories = ['Category A', 'Category B', 'Category C']
values1 = [10, 15, 20] # Values for group 1
values2 = [12, 17, 15] # Values for group 2
```



```
bar_width = 0.35

x = np.arange(len(categories))

plt.bar(x - bar_width/2, values1, bar_width, label='Group 1')
plt.bar(x + bar_width/2, values2, bar_width, label='Group 2')

plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Grouped Bar Graph')
plt.xticks(x, categories)
plt.legend()

plt.tight_layout()
plt.show()
```