



COMPUTER SCIENCE - 083

JOURNAL

Name of Candidate: Tanuj Bahar Shah
Roll No. of Candidate: 12110

Class: XII A CBSE



Certificate

Department of Information Technology
Global Indian International School SMART
Campus - Punggol, Singapore

This is to certify that this practical file is the bona fide work of
Tanuj Shah, Roll No.
12135, undertaken in partial fulfillment of the requirements of the Computer Science practical curriculum for the A.I.S.S.C.E. 2025–2026, as prescribed by the Central Board of Secondary Education (C.B.S.E.). The work has been duly completed within the stipulated time under my supervision and guidance.

Internal Examiner Signature External Examiner
Signature

Internal Examiner Number External Examiner
Number

INDEX

Sr No	Category	Page
1	Revision Tour	4
2	Functions	9
3	Modules	17
4	File Handling	23
5	Stack Operations	41
6	Practical Exam Practice	45
7	Mini Project	88
8	SQL	92
9	SQL Connector	108

Date: 23-12-2025

Revision tour

Topic	Revision Tour
Program 1	Write a program to display ASCII code of a character and vice versa.
Source Code	<pre># Program to display the ASCII code of a character char = input("Enter a character: ") ascii = ord(char) print("The ASCII code is:", ascii)</pre>
Output	<pre>/usr/local/bin/python3 "/Users/tanuj@MacBook-Air: ~ 12th Sem1 % /usr/ Enter a character: a The ASCII code is: 97 tanuj@MacBook-Air: ~ 12th Sem1 %</pre>

Topic	Revision Tour
Program 2	Write a program to input a character and to print whether a given character is an alphabet, digit or any other character.
Source Code	<pre>char = input("Enter a character: ") if char.isalpha(): print("It is an alphabetic character.") elif char.isdigit(): print("It is a digit.") else: print("It is a special character.")</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/local/ Enter a character: @ It is a special character. tanuj@MacBook-Air 12th Sem1 % /usr/local/ Enter a character: 1 It is a digit. tanuj@MacBook-Air 12th Sem1 % /usr/local/ Enter a character: a It is an alphabetic character. tanuj@MacBook-Air 12th Sem1 %</pre>

Topic	Revision Tour
Program Q7.	<p>Write a program that reads an integer N from the keyboard computes and displays the sum of the numbers from N to $(2 * N)$ if N is nonnegative. If N is a negative number, then it's the sum of the numbers from $(2 * N)$ to N. The starting and ending points are included in the sum.</p>
Source Code	<pre># Q7 sum = 0 n = int(input("Enter a number: ")) if n < 0: for i in range ((2*n)+1, n): sum+=i else: for i in range (n, (2*n)+1): sum+=i print("The sum is:", sum)</pre>
Output	<pre>Enter an alphabetic character. tanuj@MacBook-Air 12th Sem1 % /usr/lo " Enter a number: 3 The sum is: 18</pre>

Topic	Revision Tour
Program Q8.	Write a program that reads a date as an integer in the format MMDDYYYY. The program will call a function that prints out the date in the format <Month Name><day>,<year>.
Source Code	<pre># Q8 date = str(input("Enter the date in MMDDYYYY: ")) if len(date) != 8: print("Invalid date format. Please enter in MMDDYYYY format.") else: month = date[0:2] day = date[2:4] year = date[4:8] if month == "01": month_name = "January" elif month == "02": month_name = "February" elif month == "03": month_name = "March" elif month == "04": month_name = "April" elif month == "05": month_name = "May" elif month == "06": month_name = "June" elif month == "07": month_name = "July" elif month == "08": month_name = "August" elif month == "09": month_name = "September" elif month == "10": month_name = "October" elif month == "11": month_name = "November" elif month == "12": month_name = "December" else: month_name = "Invalid month" print(month_name, day, ", ", year)</pre>
Output	<pre>Enter the date in MMDDYYYY: 07132025 July 13 , 2025</pre>

--	--

Topic	Revision Tour
Program Q11.	Write a program that reads two times in military format (0900, 1730) and prints the number of hours and minutes between the two times.

Source Code	<pre># Q11 time1 = int(input("Enter the first time in HHMM format: ")) time2 = int(input("Enter the second time in HHMM format: ")) h1, m1 = divmod(time1, 100) h2, m2 = divmod(time2, 100) minutes1 = h1 * 60 + m1 minutes2 = h2 * 60 + m2 diff_minutes = abs(minutes2 - minutes1) diff_hours = diff_minutes // 60 diff_mins = diff_minutes % 60 print(diff_hours, "hours", diff_mins, "minutes")</pre>
Output	<pre>Enter the first time in HHMM format: 1013 Enter the second time in HHMM format: 1513 5 hours 0 minutes tanuj@MacBook-Air 12th Sem1 %</pre>

Functions

Topic	Functions
Program 1	Write a python function $\sin(x,n)$ to calculate the value of $\sin(x)$ using its taylor series expansion up to n terms.

	$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots,$
Source Code	<pre># Calculate sin(x) using Taylor series expansion def factorial(num): factorial = 1 if num < 0: return 0 elif num == 0: factorial=1 else: for i in range(1,num + 1): factorial = factorial*i return factorial def sin_taylor(x, n): number = 0 if n == 1: number = x else: for i in range(n): sign = (-1)**i number += (sign * (x**((2*i) + 1))) / factorial(2*i + 1) return number x = float(input("Enter the value of x (in radians): ")) n = int(input("Enter the number of terms in Taylor series: ")) if n <= 0: print("Number of terms must be a positive integer.") else: result = sin_taylor(x, n) print(f"The sine of {x} using Taylor series with {n} terms is: {result}")</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python3 /Users/tanuj/Documents Enter the value of x (in radians): 2 Enter the number of terms in Taylor series: 3 The sine of 2.0 using Taylor series with 3 terms is: 0.9333333333333333 tanuj@MacBook-Air 12th Sem1 %</pre>

Topic	Functions
Program 2	Program to perform Linear Search.
Source Code	<pre>def linear_Search(list, element): for i in range(len(list)): if list[i] == element: return i return -1 list = [1, 2, 3, 4, 5] element = 3 result = linear_Search(list, element) if result != -1: print("Element",element,"found at index",result) else: print("Element",element,"not found in the list.")</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python3.7 myprog.py Element 3 found at index 2 tanuj@MacBook-Air 12th Sem1 % []</pre>

Topic	Functions
Program 6	Create a menu driven calculator to perform arithmetic functions processing to be done in different function styles. The program runs until the user wants.

Source Code

```
def add(a, b):
    return a + b

def sub(a, b):
    return a - b

def mul(a, b):
    return a * b

def div(a, b):
    if b == 0:
        return "Error: Division by zero"
    return a / b

while True:
    print("\nMenu:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")
    print("5. Exit")
    choice = input("Enter your choice (1-5): ")
    if choice == '5':
        print("Exiting calculator. Goodbye!")
        break
    if choice in ['1', '2', '3', '4']:
        try:
            num1 = float(input("Enter first number: "))
            num2 = float(input("Enter second number: "))
        except ValueError:
            print("Invalid input. Please enter numbers.")
            continue

        if choice == '1':
            print("Result:", add(num1, num2))
        elif choice == '2':
            print("Result:", sub(num1, num2))
        elif choice == '3':
            print("Result:", mul(num1, num2))
        elif choice == '4':
            print("Result:", div(num1, num2))
    else:
        print("Invalid choice. Please select from 1 to 5.")
```

Output

```
Element 5 found at index 2
tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python
```

```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 1
Enter first number: 10
Enter second number: 15
Result: 25.0
```

```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 2
Enter first number: 15
Enter second number: 12
Result: 3.0
```

```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 3
Enter first number: 12
Enter second number: 3
Result: 36.0
```

```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 4
Enter first number: 12
Enter second number: 10
Result: 1.2
```

```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 5
Exiting calculator. Goodbye!
tanuj@MacBook-Air 12th Sem1 %
```

Topic	Functions
Program 7	<p>Write a program that accepts 3 integers from user pass them to function and return sum of values to main</p> <p>Main to pass sum to other function that calculates average and returns to main</p> <p>Main to print sum and average.</p>
Source Code	<pre> def sum(a,b,c): return a + b + c def avg(sum): return sum / 3 a = int(input("Enter first number: ")) b = int(input("Enter second number: ")) c = int(input("Enter third number: ")) total = sum(a, b, c) average = avg(total) print("Sum:", total) print("Average:", average) </pre>
Output	<pre> tanuj@MacBook-Air 12th Sem1 % /usr Enter first number: 1 Enter second number: 2 Enter third number: 3 Sum: 6 Average: 2.0 tanuj@MacBook-Air 12th Sem1 % </pre>

Topic	Functions
Program 8	Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
Source Code	<pre>ons > q8.py > ... import random # Dice def roll_dice(): return random.randint(1, 6) print(roll_dice())</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/ 4 tanuj@MacBook-Air 12th Sem1 % []</pre>

Modules

Topic	Modules
Program 1	<p>Write a program to create a library in python and import it in a program.</p> <p>Library contains</p> <ul style="list-style-type: none"> Area_rect.py with function rectangle, Area_square.py with function square, Area_Tri.py with function triangle <p>The program displays a menu to the user and calculates areas accordingly.</p>
Source Code	<pre>modules > q1 > main.py > ... 1 import area_rect 2 import area_tri 3 import area_square 4 5 while True: 6 print("\nMenu:") 7 print("1. Area of Rectangle") 8 print("2. Area of Triangle") 9 print("3. Area of Square") 10 print("4. Exit") 11 choice = input("Enter your choice (1-4): ") 12 13 if choice == '4': 14 print("Exiting program. Goodbye!") 15 break 16 17 if choice == '1': 18 length = int(input("Enter length: ")) 19 width = int(input("Enter width: ")) 20 print("Area of Rectangle:", area_rect.rectangle(length, width)) 21 22 elif choice == '2': 23 base = int(input("Enter base: ")) 24 height = int(input("Enter height: ")) 25 print("Area of Triangle:", area_tri.triangle(base, height)) 26 27 elif choice == '3': 28 side = int(input("Enter side length: ")) 29 print("Area of Square:", area_square.square(side)) 30 31 else: 32 print("Invalid choice. Please select from 1 to 4.") 33</pre> <p>modules > q1 > area_rect.py > rectangle</p> <pre>1 def rectangle(length, width): 2 return length * width</pre> <p>modules > q1 > area_square.py > square</p> <pre>1 def square(side): 2 return side * side</pre>



Global Indian
International
School

```
modules > q1 > area_tri.py > triangle
1   def triangle(base, height):
2       return 0.5 * base * height
```

Output

```
tanuj@MacBook-Air 12th Sem1 % /usr/local

Menu:
1. Area of Rectangle
2. Area of Triangle
3. Area of Square
4. Exit
Enter your choice (1-4): 1
Enter length: 12
Enter width: 123
Area of Rectangle: 1476

Menu:
1. Area of Rectangle
2. Area of Triangle
3. Area of Square
4. Exit
Enter your choice (1-4): 2
Enter base: 23
Enter height: 34
Area of Triangle: 391.0

Menu:
1. Area of Rectangle
2. Area of Triangle
3. Area of Square
4. Exit
Enter your choice (1-4): 3
Enter side length: 1
Area of Square: 1

Menu:
1. Area of Rectangle
2. Area of Triangle
3. Area of Square
4. Exit
Enter your choice (1-4): 4
Exiting program. Goodbye!
tanuj@MacBook-Air 12th Sem1 % ]
```



Topic	Modules
Program 2	<p>Write a module that has two functions.</p> <ol style="list-style-type: none"> 1) The first function takes in string as a parameter and then prints the element one by one. (easy) 2) The second function arranges the string according to the ASCII values of the Characters. (hard) 3) Add "docstrings" to the module <p>Call the module from a file placed in the same directory.</p>
Source Code	<pre>modules > q2 > 🗂 main.py > ... 1 import module 2 string = input("Enter a string: ") 3 module.split_string(string) 4 module.ascii_arrange(string) 5 </pre> <hr/> <pre>modules > q2 > 🗂 module.py > ⚡ split_string 1 """ 2 A module that provides functions to manipulate strings. 3 4 This module contains two functions: 5 1. `split_string`: Prints each character of a string on a new line. 6 2. `ascii_arrange`: Sorts the characters of a string in ascending order and prints them. 7 """ 8 9 def split_string(string): 10 """ 11 Prints each character of the string on a new line. 12 """ 13 Arguments: 14 string -- The input string to be split. 15 """ 16 for i in string: 17 print(i) 18 return 19 20 def ascii_arrange(string): 21 """ 22 Sorts the characters of the string in ascending order and prints them. 23 """ 24 Arguments: 25 string -- The input string to be sorted and printed. 26 """ 27 sorted_string = sorted(string) 28 for i in sorted_string: 29 print(i, end=' ') 30 return</pre>

Output	<pre>Enter a string: 123teststring 1 2 3 t e s t s t r i n g 1 2 3 e g i n r s s t t t % tanuj@MacBook-Air 12th Sem1 %</pre>
--------	--

Topic	File Handling
Program 1	Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file
Source Code	<pre>file handling > q1.py > ... 1 with open('files/file.txt', 'r') as file: 2 content = file.read() 3 vowels = 'aeiouAEIOU' 4 consonants = 'bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ' 5 lowercase = 'abcdefghijklmnopqrstuvwxyz' 6 uppercase = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' 7 (variable) number_of_uppercase: Literal[0] 8 number_of_uppercase = 0 9 number_of_consonants = 0 10 number_of_vowels = 0 11 for char in content: 12 if char in lowercase: 13 number_of_lowercase += 1 14 for char in content: 15 if char in uppercase: 16 number_of_uppercase += 1 17 for char in content: 18 if char in consonants: 19 number_of_consonants += 1 20 for char in content: 21 if char in vowels: 22 number_of_vowels += 1 23 print("Number of lowercase letters: ", number_of_lowercase) 24 print("Number of uppercase letters: ", number_of_uppercase) 25 print("Number of consonants: ", number_of_consonants) 26 print("Number of vowels: ", number_of_vowels)</pre>
Output	<pre>tanuj@MacBook-Air ~ % cd /Users/tanuj tanuj@MacBook-Air ~ % python q1.py Number of lowercase letters: 63 Number of uppercase letters: 6 Number of consonants: 43 Number of vowels: 26 tanuj@MacBook-Air ~ %</pre>

File Handling

Topic	File Handling
Program 2	Read a text file line by line and display each word separated by a #.
Source Code	<pre>with open('file handling/files/file.txt', 'r') as file: for line in file: words = line.strip().split('#') for i in words: print(i, end=' ') print() # Print a newline after each line of words</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/local/ This is a sample text file. Words are split using hashtags Created By Tanuj Shah . a a a a tanuj@MacBook-Air 12th Sem1 %</pre>

Topic	File Handling
-------	---------------



Program 3	Write a program to count the number of words in a file.
Source Code	<pre>handling > ✎ q3.py > ... 1 with open('file handling/files/file.txt', 'r') as file: 2 content = file.read() 3 words = content.split() 4 word_count = len(words) 5 print("Number of words in the file:", word_count)</pre>
Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/ Number of words in the file: 12 tanuj@MacBook-Air 12th Sem1 % █</pre>

Topic	File Handling
Program 4	Write a program to write those lines which have the character 'p' from one text file to another text file.

Source Code	<pre>file handling > q4.py > ... 1 file1 = open('file handling/files/file.txt', 'r') 2 file2 = open('file handling/files/file2.txt', 'w') 3 for line in file1: 4 if 'p' in line: 5 file2.write(line) 6 file1.close() 7 file2.close()</pre>
Output	<pre>file handling > files > - mez.txt This is a sample text file.#Words#are #split#using#hashtags</pre>

Topic	File Handling
Program 5	Write a program to find the most common words in a file.

Source Code

```
with open('file handling/files/file.txt', 'r') as file:
    content = file.read()
    words = content.split()
    word_counts = {}
    for word in words:
        word = word.lower()
        if word in word_counts:
            word_counts[word] += 1
        else:
            word_counts[word] = 1
    highest_word = max(word_counts, key=word_counts.get)
    highest_count = word_counts[highest_word]
    print("The word", highest_word, "appears the most with a count of", highest_count)
```

Output

```
tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python
The word a appears the most with a count of 5
tanuj@MacBook-Air 12th Sem1 %
```

Topic	File Handling
Program 6	Write a program to perform read and write operations with a .csv file.

Source Code	<pre> import csv with open('file handling/files/data.csv', 'w+', newline='') as file: writer = csv.writer(file) writer.writerow(['Title1', 'Title2', 'Title3']) writer.writerow(['Content1', 'Content2', 'Content3']) writer.writerow(['Content4', 'Content5', 'Content6']) writer.writerow(['Content7', 'Content8', 'Content9']) reader = csv.reader(file) file.seek(0) # Move the cursor back to the beginning of the file for row in reader: print(row) </pre>
Output	<pre> tanuj@MacBook-Air 12th Sem1 % /usr/local ['Title1', 'Title2', 'Title3'] ['Content1', 'Content2', 'Content3'] ['Content4', 'Content5', 'Content6'] ['Content7', 'Content8', 'Content9'] tanuj@MacBook-Air 12th Sem1 % </pre>

Topic	File Handling
Program 7	<p>Remove all the lines that contain the character 'a' in a file and write it to another file</p> <p>Create a binary file with name and roll number. Search for a given roll</p>

	number and display the name, if not found display appropriate message.
Source Code	<pre> import pickle fin = open('file handling/files/file.txt', 'r') fout = open('file handling/files/output.txt', 'w') for line in fin: if 'a' not in line: fout.write(line) fin.close() fout.close() records = [{'name': 'Tanuj', 'roll': 101}, {'name': 'Test', 'roll': 102}, {'name': 'Usernotfound', 'roll': 103}] with open("file handling/files/file.dat", 'wb') as f: for record in records: pickle.dump(record, f) def search_roll_number(filename, roll_no): found = False with open(filename, 'rb') as f: try: while True: record = pickle.load(f) if record['roll'] == roll_no: print("Name:", record['name']) found = True break except EOFError: pass if not found: print("Roll number not found.") search_roll_number("file handling/files/file.dat", 102) </pre>
Output	<ul style="list-style-type: none"> tanuj@MacBook-Air 12th Sem1 % /usr Name: Test tanuj@MacBook-Air 12th Sem1 %

Topic	File Handling
-------	---------------



Program 8

Create a binary file with roll number, name and marks. Input a roll number and update the marks.

Source Code

```

filename > = qd.py > ...
import pickle

FILENAME = 'file handling/files/students.dat'

def create_file():
    n = int(input("Enter number of students: "))
    records = []
    for _ in range(n):
        roll = int(input("Enter roll number: "))
        name = input("Enter name: ")
        marks = int(input("Enter marks: "))
        records.append({'roll': roll, 'name': name, 'marks': marks})
    with open(FILENAME, 'wb') as f:
        pickle.dump(records, f)
    print("File created successfully.")

def update_marks():
    roll_no = int(input("Enter roll number to update marks: "))
    found = False
    try:
        with open(FILENAME, 'rb') as f:
            records = pickle.load(f)
        for rec in records:
            if rec['roll'] == roll_no:
                print(f"Current marks: {rec['marks']} ")
                rec['marks'] = int(input("Enter new marks: "))
                found = True
                break
        if found:
            with open(FILENAME, 'wb') as f:
                pickle.dump(records, f)
            print("Marks updated successfully.")
        else:
            print("Roll number not found.")
    except FileNotFoundError:
        print("File not found. Please create the file first.")

while True:
    print("\n1. Create File\n2. Update Marks\n3. Exit")
    choice = input("Enter choice: ")
    if choice == '1':
        create_file()
    elif choice == '2':
        update_marks()
    elif choice == '3':
        break
    else:
        print("Invalid choice.")

```

Output	<pre> tanuj@MacBook-Air 12th Sem1 % /usr/local/b 1. Create File 2. Update Marks 3. Exit Enter choice: 1 Enter number of students: 2 Enter roll number: 1 Enter name: Tanuj Enter marks: 0 Enter roll number: 2 Enter name: Test Enter marks: 100 File created successfully. 1. Create File 2. Update Marks 3. Exit Enter choice: 2 Enter roll number to update marks: 1 Current marks: 0 Enter new marks: 1 Marks updated successfully. 1. Create File 2. Update Marks 3. Exit Enter choice: 3 tanuj@MacBook-Air 12th Sem1 % </pre>
--------	--

Topic	File Handling
Program 9	Create a CSV file by entering user-id and password, read and search the password for the given user id.

Source Code	<pre> import csv def write_csv(): with open('file handling/files/file.csv', 'w') as file: writer = csv.writer(file) user_id = input("Enter user ID: ") password = input("Enter password: ") writer.writerow([user_id, password]) return def read_csv(user_id): with open('file handling/files/file.csv', 'r') as file: reader = csv.reader(file) for row in reader: if row[0] == user_id: print("Password for user ID", user_id, "is:", row[1]) return print("User ID not found.") return while True: print("1. Write to CSV\n2. Read from CSV\n3. Exit") choice = input("Enter your choice: ") if choice == '1': write_csv() elif choice == '2': user_id = input("Enter user ID to read password: ") read_csv(user_id) elif choice == '3': break else: print("Invalid choice, please try again.") </pre>
--------------------	---

Output	<pre>tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python3 1. Write to CSV 2. Read from CSV 3. Exit Enter your choice: 1 Enter user ID: 1 Enter password: 2 1. Write to CSV 2. Read from CSV 3. Exit Enter your choice: 2 Enter user ID to read password: 1 Password for user ID 1 is: 2 1. Write to CSV 2. Read from CSV 3. Exit Enter your choice: 3 tanuj@MacBook-Air 12th Sem1 %</pre>
--------	---

Topic	File Handling
-------	---------------

Program 10	<p>Write code to do following</p> <ol style="list-style-type: none"> 1. Create an Excel file (save as csv) 2. Import file created in part 1 in Python file and display content of csv 3. Append content to files created in part 1 through python. file (read values from user) 4. Display updated csv file from python file
Source Code	<pre>cdmg > q10.py > ... import csv def display_content(): with open('file handling/files/file.csv', 'r') as file: reader = csv.reader(file) for row in reader: print(row) def append_to_csv(): with open('file handling/files/file.csv', 'a', newline='') as file: writer = csv.writer(file) user_id = input("Enter user ID: ") password = input("Enter password: ") writer.writerow([user_id, password]) print("Data appended successfully.") display_content() while True: print("1. Append to CSV\n2. Display Content\n3. Exit") choice = input("Enter your choice: ") if choice == '1': append_to_csv() elif choice == '2': display_content() elif choice == '3': break else: print("Invalid choice, please try again.")</pre>

Output

```
tanuj@MacBook-Air 12th Sem1 % /usr/lo
1. Append to CSV
2. Display Content
3. Exit
Enter your choice: 1
Enter user ID: 3
Enter password: 4
Data appended successfully.
['1', '2']
1. Append to CSV
2. Display Content
3. Exit
Enter your choice: 2
['1', '2']
['3', '4']
1. Append to CSV
2. Display Content
3. Exit
Enter your choice: █
```

Program 11	<p>Appending content to the file</p> <ol style="list-style-type: none"> 1. create a file with 'w' 2. print original file by reading with 'r' 3. add content to file with 'a' 4. reprint file to show updated content
Source Code	<pre>handling > q11.py > ... def create_file(): with open('file handling/files/file.txt', 'w') as file: return def write_to_file(data): with open('file handling/files/file.txt', 'a') as file: file.write(data + '\n') return def read_file(): with open('file handling/files/file.txt', 'r') as file: content = file.readlines() for line in content: print(line.strip()) return while True: print("1. Create File\n2. Write to File\n3. Read File\n4. Exit") choice = input("Enter your choice: ") if choice == '1': create_file() print("File created successfully.") elif choice == '2': data = input("Enter data to write to file: ") write_to_file(data) print("Data written successfully.") elif choice == '3': print("File content:") read_file() elif choice == '4': break else: print("Invalid choice, please try again.")</pre>

Output

```
tanuj@MacBook-Air 12th Sem1 % /usr/local/bin/python3 "/Users/  
1. Create File  
2. Write to File  
3. Read File  
4. Exit  
Enter your choice: 1  
File created successfully.  
1. Create File  
2. Write to File  
3. Read File  
4. Exit  
Enter your choice: 2  
Enter data to write to file: ABCD\n/n\\n/n/nEFGH  
Data written successfully.  
1. Create File  
2. Write to File  
3. Read File  
4. Exit  
Enter your choice: 3  
File content:  
ABCD\n/n\\n/n/nEFGH  
1. Create File  
2. Write to File  
3. Read File  
4. Exit  
Enter your choice: 4  
tanuj@MacBook-Air 12th Sem1 %
```

Topic

Functions

Program 12	<p>Write a Python program to</p> <ol style="list-style-type: none">1. Create a binary file (read Item_code, Item_Name, Item_Price, Item_Qty from user and write to file) write as many objects as user wants2. Read from binary file (display contents of file created in part 1)3. Search for a particular Item_code in this file and print if it exists or not4. Modify record of a particular user entered Item_code (use seek and tell pointers to set pointer position)
------------	--

Source Code

```

file handling > q12.py > ...
1  import pickle
2  file_location = 'file handling/files/items.dat'
3  def create_file(file_location):
4      with open(file_location, 'wb') as file:
5          items = []
6          n = int(input("Enter number of items: "))
7          for _ in range(n):
8              item_code = input("Enter item code: ")
9              item_name = input("Enter item name: ")
10             item_price = float(input("Enter item price: "))
11             item_quantity = int(input("Enter item quantity: "))
12             items.append({'code':item_code, 'name': item_name, 'price': item_price, 'quantity': item_quantity})
13             pickle.dump(items, file)
14             print("File created successfully.")
15             return
16  def display_items(file_location):
17      try:
18          with open(file_location, 'rb') as file:
19              items = pickle.load(file)
20              if not items:
21                  print("No items found.")
22                  return
23                  print("Items in the file:")
24                  for item in items:
25                      print("Code: {}, Name: {}, Price: {}, Quantity: {}".format(item['code'], item['name'], item['price'], item['quantity']))
26      except FileNotFoundError:
27          print("File not found. Please create the file first.")
28          return
29  def search_item(file_location):
30      try:
31          with open(file_location, 'rb') as file:
32              items = pickle.load(file)
33              search_name = input("Enter item code to search: ")
34              found = False
35              for item in items:
36                  if item['code'].lower() == search_name.lower():
37                      print("Item found: Name: {}, Price: {}, Quantity: {}".format(item['name'], item['price'], item['quantity']))
38                      found = True
39                      break
40              if not found:
41                  print("Item not found.")
42      except FileNotFoundError:
43          print("File not found. Please create the file first.")
44          return
45  def modify_record(file_location):
46      try:
47          with open(file_location, 'rb') as file:
48              items = pickle.load(file)
49              search_code = input("Enter item code to modify: ")
50              found = False
51              for item in items:
52                  if item['code'].lower() == search_code.lower():
53                      item['name'] = input("Enter new item name: ")
54                      item['price'] = float(input("Enter new item price: "))
55                      item['quantity'] = int(input("Enter new item quantity: "))
56                      found = True
57                      break
58              if not found:
59                  print("Item not found.")
60                  return
61              with open(file_location, 'wb') as file:
62                  pickle.dump(items, file)
63                  print("Record modified successfully.")
64      except FileNotFoundError:
65          print("File not found. Please create the file first.")
66          return
67  while True:
68      print("\nMenu:")
69      print("1. Create file")
70      print("2. Display items")
71      print("3. Search item")
72      print("4. Modify record")
73      print("5. Exit")
74      choice = input("Enter your choice: ")
75
76      if choice == '1':
77          create_file(file_location)
78      elif choice == '2':
79          display_items(file_location)
80      elif choice == '3':
81          search_item(file_location)
82      elif choice == '4':
83          modify_record(file_location)
84      elif choice == '5':
85          print("Exiting the program.")
86          break
87      else:
88          print("Invalid choice, please try again.")

```

Output

```

Menu:
1. Create file
2. Display items
3. Search item
4. Modify record
5. Exit
Enter your choice: 1
Enter number of items: 2
Enter item code: AB
Enter item name: Pencil
Enter item price: 20
Enter item quantity: 30
Enter item code: BC
Enter item name: Erase
Enter item price: 10
Enter item quantity: 32
File created successfully.

Menu:
1. Create file
2. Display items
3. Search item
4. Modify record
5. Exit
Enter your choice: 2
Items in the file:
Code: AB, Name: Pencil, Price: 20.0, Quantity: 30
Code: BC, Name: Erase, Price: 10.0, Quantity: 32

Menu:
1. Create file
2. Display items
3. Search item
4. Modify record
5. Exit
Enter your choice: 3
Enter item code to search: AB
Item found: Name: Pencil, Price: 20.0, Quantity: 30

Menu:
1. Create file
2. Display items
3. Search item
4. Modify record
5. Exit
Enter your choice: 4
Enter item code to modify: BC
Enter new item name: Eraser
Enter new item price: 12
Enter new item quantity: 33
Record modified successfully.

Menu:
1. Create file
2. Display items
3. Search item
4. Modify record
5. Exit
Enter your choice: 2
Items in the file:
Code: AB, Name: Pencil, Price: 20.0, Quantity: 30
Code: BC, Name: Eraser, Price: 12.0, Quantity: 33

```

Stacks

Topic	Stacks
Program 1	<p>Write a menu driven program to show implementation of all stack operations.</p> <p>1.peek 2.push 3.pop 4.display</p> <p>Program to run until user wants</p>

Source Code

```
stack = []
position = -1
def push(item):
    global position
    stack.append(item)
    position += 1
    print(f"Item '{item}' pushed onto stack.")
def pop():
    global position
    if position == -1:
        print("Stack is empty. Cannot pop.")
        return None
    item = stack.pop()
    position -= 1
    print(f"Item '{item}' popped from stack.")
    return item
def peek():
    if position == -1:
        print("Stack is empty. Nothing to peek.")
        return None
    item = stack[position]
    print(f"Item at top of stack: '{item}'")
    return item
def display():
    if position == -1:
        print("Stack is empty.")
        return
    print("Current stack items:")
    for i in range(position + 1):
        print(f"Position {i}: {stack[i]}")
while True:
    print("\nStack Operations:")
    print("1. Push item")
    print("2. Pop item")
    print("3. Peek at top item")
    print("4. Display stack")
    print("5. Exit")
    choice = input("Enter your choice: ")

    if choice == '1':
        item = input("Enter item to push: ")
        push(item)
    elif choice == '2':
        pop()
    elif choice == '3':
        peek()
    elif choice == '4':
        display()
    elif choice == '5':
        print("Exiting stack operations.")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output

```
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 1  
Enter item to push: a  
Item 'a' pushed onto stack.  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 3  
Item at top of stack: 'a'  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 1  
Enter item to push: b  
Item 'b' pushed onto stack.  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 3  
Item at top of stack: 'b'  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 4  
Current stack items:  
Position 0: a  
Position 1: b  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 2  
Item 'b' popped from stack.  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 4  
Current stack items:  
Position 0: a  
  
Stack Operations:  
1. Push item  
2. Pop item  
3. Peek at top item  
4. Display stack  
5. Exit  
Enter your choice: 5  
Exiting stack operations.  
tanuj@MacBook-Air 12th Sem1 % █
```



Python practical exam practice

Topic	File Handling
Program 1	<p>Write an interactive menu driven program to create a text file and then display the file. Create another text file by converting each line of the newly created text file into an uppercase string. Display the newly created file.</p>
Source Code	<pre>file into an uppercase string. Display the newly created file.'''\nchoice = 0\nwhile True:\n choice = int(input("1 To Create a text file and displaying it\n2 To convert each line of the newly created text file into an uppercase string\n3 To Display the newly created file\nPlease Enter choice: "))\n if choice == 1:\n file = open('cfile.txt', 'w')\n text = input("Enter text to be written: ") \n file.write(text)\n file.close()\n data = file.read()\n file.close()\n print("file data: ") \n print(data)\n elif choice == 2:\n file = open('cfile.txt', 'r+') data = file.read() file.truncate() file.write(data.upper()) file.close() print(data) elif choice == 3: file = open('cfile.txt', 'r') data = file.read() file.close() print(data)\n print(data)</pre>
Output	<p>PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS</p> <pre>tanuj@MacBook-Air: ~ Assignment % /usr/local/bin/python3 "/Users/tanuj/Documents/School/CS Assignments/12th Sem2/Assignment/tshah cs practical.py" 1 To Create a text file and displaying it 2 To convert each line of the newly created text file into an uppercase string 3 To Display the newly created file Please Enter choice: 1</pre>

Topic	File Handling
Program 2	<p>Declare a dictionary named telerec, containing name and telephone number. A binary data file “Tele.dat” stores data of the type telerec. Write functions to do following.</p> <p>1. To append records in the file</p> <p>Display the name for a given telephone number, if the telephone number doesn’t exist then display error message “Record not found”.</p>

Source Code

```

display the name for a given telephone number,
exist then display error message "Record not found"
import pickle
telrec={}
def addrec(name,phone, obj):
    obj[phone] = name
def createfile():
    file = open('tele.dat', 'wb')
    pickle.dump(telrec, file)
    file.close()
def append(phone,name):
    file = open('tele.dat', 'rb')
    data = pickle.load(file)
    addrec(phone, name, data)
    file.close()
    file = open('tele.dat', 'wb')
    pickle.dump(data, file)
    file.close()
def checkr(phone):
    file = open('tele.dat', 'rb')
    data = pickle.load(file)
    if data[phone]:
        print(data[phone])
    else:
        raise(NameError,"Record not found")
createfile()
append('test', 32)
checkr(32)

```

Output

```

tanuj@MacBook-Air Assignment % /usr/local/bin/python3 "/Users/tanuj/Downloads/Assignment 3.py"
test

```

Topic	File Handling
Program 3	<p>Write a python program to get 5 items details (itemno, name, price, category) from the user and create menu driven program using functions to</p> <ol style="list-style-type: none"> 1. create a csv file for above item details and display csv file content 2. create a binary file for above item details and display binary file content
Source Code	<pre> 2. create a binary file for above item details and display binary file content""" import csv import pickle obj = [] for i in range (0,5): itemno = input("Enter item number: ") name = input("Enter name: ") price = input("Enter price: ") category = input("Enter category: ") obj.append([itemno, name, price, category]) choice = 0 while True: choice = int(input("Enter choice 1 to create csv file and display it\n2 to create binary file.")) if choice == 1: file = open("cscsvfile.csv", "w") writer = csv.writer(file) writer.writerows(obj) file.close() file = open("cscsvfile.csv", "r") reader = csv.reader(file) print(reader.readlines()) file.close() if choice == 2: file = open("cpsuedocsvdatfile.dat", "wb") pickle.dump(obj, file) file.close() file = open("cpsuedocsvdatfile.dat", "rb") data = pickle.load(file) file.close() print(data) """ !!!Qd. Write a program to read a text file and create a dictionary that stores the Enter choice 1 to create csv file and display it 2 to create binary file. [{'1': 'Eraser', '10': 'Stationery'}, {'2': 'Mobile', '40000': 'Electronics'}, {'3': 'Pizza', '120': 'Food'}, {'2': 'Burger', '100': 'Food'}, {'3434': 'Chocolate', '5': 'Confectionary'}] </pre>
Output	

Topic	Dictionary
Program 4	<p>Write a program to read a text file and create a dictionary that stores the frequency table of alphabetic characters and digits. Print dictionary.</p>
Source Code	<pre>'''Q4 Write a program to read a text file and create a dictionary that stores the frequency table of alphabetic characters and digits. Print dictionary.''' file = open('csfile.txt', 'r') text = file.read() dic = {} for i in text: if i in dic: dic[i]+=1 else: dic[i]= 1 print(dic)</pre>
Output	<pre>tanuj@MacBook-Air: Assignment % /usr/local/bin/python3 "/Users/tanuj/Documents/School/CS Assignments/12th Sem2/Assignment/tshah cs practice" {'m': 2, 'e': 3, 'o': 4, 'w': 1, ' ': 6, 'h': 1, 'i': 2, 'b': 1, 'y': 2, 'l': 3, 'v': 1, 'u': 2, 'a': 2, 'r': 1, 'n': 1}</pre>

Topic	File Handling
Program 5	<p>Write a python program to get 5 items details (itemno, name, price, category) from the user and create a csv Items.csv, create another file highitems.csv, containing only those item details from Items.csv where price>250.</p>
Source Code	<pre> '''Q5 Write a python program to get 5 items details (itemno, name, price, category) from the user and create a csv Items.csv, create another file highitems.csv, containing only those item details from Items.csv where price>250.''' import csv obj = [] for i in range (0,5): itemno = input("Enter item number: ") name = input("Enter name: ") price = int(input("Enter price: ")) category = input("Enter category: ") obj.append([itemno, name, price, category]) file = open("items.csv", "w") writer = csv.writer(file) writer.writerows(obj) file.close() file = open("items.csv", "r") reader = csv.reader(file) for i in reader: print(i) file.close() newobj=[] for i in obj: if i[2] > 250: newobj.append(i) file = open("highitems.csv", "w") writer = csv.writer(file) writer.writerows(newobj) file.close() file = open("highitems.csv", "r") reader = csv.reader(file) for i in reader: print(i) file.close()</pre>

Output

```
tanuj@MacBook-Air Assignment % /u
Enter name: 1
Enter price: 1
Enter category: 11
['1', '1', '1', '1']
['1', '1', '1', '1']
['1', '1', '11', '1']
['1', '1', '1', '1']
['1', '1', '1', '11']
Enter 1 for factorial
```

Topic	Functions
Program 6	<p>Write a menu driven program to perform following operations depending on user choice, program to run repeatedly with function calls until user presses ‘exit’.</p> <ol style="list-style-type: none"> 1. Factorial of number (user enters value) 2. Fibonacci series (user enters number of terms) 3. Check if a number is prime or not (user enters number) <p>exit</p>
Source Code	<pre> while True: choice = int(input("Enter 1 for factorial\nEnter 2 for fibonacci series\nEnter 3 for prime or not\n4 To break\nEnter:")) if choice == 1: result = 1 n = int(input("Number: ")) if n<0: print("No factorial") else: for i in range(1,n+1): result *= i print(result) elif choice == 2: n = int(input('Number: ')) prev_num = 0 curr_num = 1 output = [] while curr_num <= n: output.append(prev_num) prev_num, curr_num = curr_num, prev_num+curr_num print(output) elif choice == 3: n = int(input('Number: ')) prime = True for i in range(1,n): if n%i==0: prime = False print(prime) elif choice == 4: print("Exiting..") break else: print("Invalid. Exiting..") break </pre>

Output	<pre>tanuj@MacBook-Air Assignment % /usr/ Enter 1 for factorial Enter 2 for fibonacci series Enter 3 for prime or not 4 To break Enter:1 Number: 6 720 Enter 1 for factorial Enter 2 for fibonacci series Enter 3 for prime or not 4 To break Enter:2 Number: 7 [0, 1, 1, 2, 3] Enter 1 for factorial Enter 2 for fibonacci series Enter 3 for prime or not 4 To break Enter:3 Number: 67 False Enter 1 for factorial Enter 2 for fibonacci series Enter 3 for prime or not 4 To break Enter:■</pre>
--------	--

Topic	Stack
Program 7	<p>NUM is a list containing 10 integers. Create a program with separate user defined functions to perform the following operations based on this list.</p> <ul style="list-style-type: none"> Traverse the content of the list and push the even numbers into a stack. <p>For Example: If the sample Content of the list is as follows: NUM=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38] Sample Output of the code should be: 38 22 98 56 34 12 </p>
Source Code	<pre>num=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38] stack=[] def filter(): for i in num: if i%2==0: stack.append(i) print(stack) filter()</pre>
Output	 tanuj@MacBook-Air Assignment % /usr/local/bin/python3 stack.py [12, 34, 56, 98, 22, 38]

Topic	Stack
Program 8	<p>NUM is a list containing 10 integers. Create a program with separate user defined functions to perform the following operations based on this list. .</p> <ol style="list-style-type: none">1. Push2. Pop3. Peek4. Display5. Exit



Source Code

```
5. Exit'''  
num=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]  
stack=[]  
position=0  
def push(val):  
    global position  
    stack.insert(position,val)  
    position+=1  
def pop():  
    global position  
    val = stack.pop(position-1)  
    position-=1  
    return val  
def peek():  
    global position  
    val = stack[position-1]  
    return val  
def display():  
    global position  
    for i in range(0,position):  
        print(i, end="")  
def exit():  
    print("Exiting..")  
push(1)  
push(2)  
print(pop())  
display()  
peek()
```

Output

```
[Errno 2] No such file or directory: 'Assignment/tshah cs practical.py'@tanuj[MacBook-Air Assignment % /usr/local/bin/python3 "/Users/tanuj/Documents/School/CS Assignments/12th Sem2/Assignment/tshah cs practical.py"
```

Topic	Stack
Program 9	<p>Julie has created a dictionary containing names and marks as key value pairs of 6 students. Write a program, with separate user defined functions to perform the following operations:</p> <ul style="list-style-type: none"> • Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75. • Pop and display the content of the stack.
Source Code	<pre>Julie = {'ragu':97, 'himesh':98, 'neel':97, 'aashvan':39,'kush':96,'harman':69} stack =[] def push(item): for i in Julie: if Julie[i]>75: stack.append(i) def pop(): item = stack.pop() return item push('e') print(pop()) print(pop()) print(pop()) print(pop())</pre>
Output	<pre>tanuj@MacBook-Air Ass kush neel himesh ragu tanuj@MacBook-Air Ass</pre>

Topic	File Handling
Program 10	<p>Write a menu driven program to perform following functions on a text file depending on user choice</p> <ol style="list-style-type: none"> 1. Read first 3 lines from file (line by line) 2. Read first 30 bytes from file 3. Exit <p>Note : Raise appropriate exception if file not found</p>
Source Code	<pre># ''' text=[] try: r= open('file.txt', 'r') while True: choice = int(input("Enter choice: ")) if choice == 1: for i in range(0,3): text.append(r.readline()) print(text) if choice == 2: r.seek(0) bites = r.read(30) print(bites) if choice == 3: r.close() break except FileNotFoundError: print("FNF") '''</pre>
Output	<pre>tanuj@MacBook-Air Assignment % /usr/local/bin/python3.7 test123.py Enter choice: 1 ['hi\n', 'tanuj here\n', 'test123\n'] Enter choice: 2 hi tanuj here test123 67 Enter choice: 3</pre>



Topic	File Handling
Program 11	<p>Write a program to get student data (rollNo, name and marks) from user and write onto a binary file. The program should be able to get data from the user and write onto the file as long as the user wants.</p> <p>Add user defined function to facilitate reading object from above file and display.</p>
Source Code	<pre># ''' import pickle data=[] iterations = int(input("Enter number of values to enter: ")) for i in range(0,iterations): roll=int(input("Enter roll no: ")) name=input("Enter name: ") marks=int(input("Enter marks: ")) data.append([roll,name,marks]) with open('file.dat', 'wb') as f: pickle.dump(data,f) def read(): with open('file.dat', 'rb') as e: content=pickle.load(e) return content print(read()) # '''</pre>

Output

```
ent/tshah cs practice
Enter name: Rasi
Enter marks: 54
Enter roll no: 4
Enter name: Yoad
Enter marks: 12
Enter roll no: 5
Enter name: Dhpr
Enter marks: 0
Enter roll no: 6
Enter name: Anth
Enter marks: 13
Enter roll no: 7
Enter name: Ombi
Enter marks: 35
Enter roll no: 8
Enter name: Smir
Enter marks: 10
Enter roll no: 9
Enter name: Pigo
Enter marks: 67
Enter roll no: 10
Enter name: Defa
Enter marks: 20
```

```
Enter marks: 20
[[1, 'Namo', 20], [2, 'Amsh', 34], [3, 'Rasi', 54], [4, 'Yoad', 12], [5, 'Dhpr', 0], [6, 'Anth', 13], [7, 'Ombi', 35],
[8, 'Smir', 10], [9, 'Pigo', 67], [10, 'Defa', 20]]
```

Topic	Search
Program 12	<p>Write a Python program store list of n (user entered) integers.</p> <p>Program to take user choice and process following user defined functions</p> <ol style="list-style-type: none">1. Linear Search2. Binary Search3. Exit <p>Note: Program to run repeatedly until user wants and the processing should be traditional iterative method.</p>



Source Code

```
TRADITIONAL ITERATIVE METHODS.  
# ***  
def l_search(lst, key):  
    for i in range(len(lst)):  
        if lst[i] == key:  
            return i  
    return -1  
def b_search(lst, key):  
    low = 0  
    high = len(lst) - 1  
    while low <= high:  
        mid = (low + high) // 2  
        if lst[mid] == key:  
            return mid  
        elif lst[mid] < key:  
            low = mid + 1  
        else:  
            high = mid - 1  
    return -1  
leest = []  
n = int(input("Enter number of elements: "))  
for i in range(n):  
    num = int(input("Enter element: "))  
    leest.append(num)  
while True:  
    print("\nMenu:")  
    print("1. Linear Search")  
    print("2. Binary Search")  
    print("3. Exit")  
    choice = int(input("Enter your choice: "))  
    if choice == 1:  
        key = int(input("Enter element to search: "))  
        pos = l_search(leest, key)  
        if pos != -1:  
            print("Element found at position", pos+1)  
        else:  
            print("Element not found.")  
    elif choice == 2:  
        # Binary search needs sorted list  
        sorted_lst = sorted(leest)  
        print("Sorted List:", sorted_lst)  
        key = int(input("Enter element to search: "))  
        pos = b_search(sorted_lst, key)  
        if pos != -1:  
            print("Element found at position", pos+1)  
        else:  
            print("Element not found.")  
    elif choice == 3:  
        print("Exiting program...")  
        break  
    else:  
        print("WRONG!!")
```

Output

```
tanuj@MacBook-Air Assignment % /  
py"  
Enter number of elements: 3  
Enter element: 10  
Enter element: 12  
Enter element: 14  
  
Menu:  
1. Linear Search  
2. Binary Search  
3. Exit  
Enter your choice: 1  
Enter element to search: 10  
Element found at position 1  
  
Menu:  
1. Linear Search  
2. Binary Search  
3. Exit  
Enter your choice: 2  
Sorted List: [10, 12, 14]  
Enter element to search: 12  
Element found at position 2  
  
Menu:  
1. Linear Search  
2. Binary Search  
3. Exit  
Enter your choice: █
```



Topic	Functions
Program 13	<p>Write a Python program store list of n (user entered) integers.</p> <p>Program to take user choice and process following user defined functions</p> <ol style="list-style-type: none">1. Arrange values in Ascending order2. Write all multiples of 5 from above list to a text file.3. Exit <p>Note: Program to run repeatedly until user wants and the processing should be traditional iterative method.</p>

Source Code

```
traditional iterative method...  
# ***  
def arr(lst):  
    n = len(lst)  
    for i in range(n - 1):  
        for j in range(n - i - 1):  
            if lst[j] > lst[j + 1]:  
                lst[j], lst[j + 1] = lst[j + 1], lst[j]  
    return lst  
def wr(lst):  
    multiples = []  
    for num in lst:  
        if num % 5 == 0:  
            multiples.append(num)  
    with open("mult.txt", "w") as f:  
        for num in multiples:  
            f.write(str(num) + "\n")  
lst = []  
n = int(input("Enter number of elements to be entered: "))  
for i in range(n):  
    num = int(input("Enter element pls: "))  
    lst.append(num)  
while True:  
    print("\nMenu Card:")  
    print("1. Arrange values in Ascending order")  
    print("2. Write all multiples of 5 to a text file")  
    print("3. Exit")  
    choice = int(input("Enter your choice: "))  
    if choice == 1:  
        lst = arr(lst)  
        print("List arranged in ascending order:", lst)  
    elif choice == 2:  
        wr(lst)  
    elif choice == 3:  
        print("Exiting the program.....")  
        break  
    else:  
        print("Choose something else pls")
```

Output

```
tanuj@MacBook-Air Assignment % /usr/local/bin/python3 "/Users/tanuj/Downloads/Assignment.py"
Enter number of elements to be entered: 5
Enter element pls: 1
Enter element pls: 2
Enter element pls: 5
Enter element pls: 10
Enter element pls: 15

Menu Card:
1. Arrange values in Ascending order
2. Write all multiples of 5 to a text file
3. Exit
Enter your choice: 1
List arranged in ascending order: [1, 2, 5, 10, 15]

Menu Card:
1. Arrange values in Ascending order
2. Write all multiples of 5 to a text file
3. Exit
Enter your choice: 2

Menu Card:
1. Arrange values in Ascending order
2. Write all multiples of 5 to a text file
3. Exit
Enter your choice: 3
Exiting the program
```

Topic	Modules
Program 14	<p>Create a package named ‘Report’ which reads student details (rollNo, Name, Marks of 5 subjects) and generates a report card (5 subjects) with help of following import modules</p> <ol style="list-style-type: none"> 1. Module “Percen” to take total marks from Report module and return percentage of same. 2. Module “Grade” to take percentage from Report module and return grades as follows >90. A >80 B >70. C >60 F 3. Report module to use “Percen” and “Grade” and generate and print report card of student
Source Code	<pre>Report > Report.py > ... 1 import percen 2 import grade 3 def generate_report(): 4 roll_no = input("Enter Roll Number: ") 5 name = input("Enter Student Name: ") 6 marks = {} 7 sub=['English', 'Math','CS','Chem','Phy'] 8 total = 0 9 print("\nEnter marks for 5 subjects pls one subject at a time:") 10 for i in sub: 11 print("Enter for", i) 12 m = float(input("Marks: ")) 13 marks[i]=m 14 total += m 15 percentage = percen.find_percentage(total) 16 gre = grade.find_grade(percentage) 17 print("THIS IS A REPORT CARD") 18 print("Roll No:",roll_no) 19 print("Name: ", name) 20 print("Marks:", marks) 21 print("Total Marks:",total) 22 print("Percentage: ", percentage) 23 print("Grade:", gre) 24 generate_report()</pre>

```
ort > ⚡ percen.py > ...
def find_percentage(total_marks):
    perce = total_marks / 5
    return perce
```

```
ort > ⚡ grade.py > ⚡ find_grade
def find_grade(percentage):
    if percentage > 90:
        return "A"
    elif percentage > 80:
        return "B"
    elif percentage > 70:
        return "C"
    elif percentage > 60:
        return "D"
    else:
        return "Fail"
```

Output	<pre>Enter Roll Number: 12 Enter Student Name: NaMo Enter marks for 5 subjects pls one subject at a time: Enter for English Marks: 13 Enter for Math Marks: 16 Enter for CS Marks: 4 Enter for Chem Marks: 24 Enter for Phy Marks: 37 THIS IS A REPORT CARD Roll No: 12 Name: NaMo Marks: {'English': 13.0, 'Math': 16.0, 'CS': 4.0, 'Chem': 24.0, 'Phy': 37.0} Total Marks: 94.0 Percentage: 18.8 Grade: Fail</pre>
--------	---

Topic	Functions
Program 15	<p>Write an interactive Python program menu driven program as per user choice to perform following options with help of user defined functions on Nested Lists.</p> <ol style="list-style-type: none"> 1. Matrix addition 2. Sum of all elements of Matrix 3. Transpose of Matrix 4. Exit
Source Code	<pre> def input_matrix(): rows = int(input("Enter number of rows: ")) cols = int(input("Enter number of columns: ")) matrix = [] print("Enter matrix elements row-wise:") for i in range(rows): row = [] for j in range(cols): val = int(input("Enter element [" + str(i+1) + "][" + str(j+1) + "]: ")) row.append(val) matrix.append(row) return matrix def display_matrix(matrix): for row in matrix: for val in row: print(val, end="\t") print() def add_matrices(m1, m2): rows = len(m1) cols = len(m1[0]) result = [] for i in range(rows): row = [] for j in range(cols): row.append(m1[i][j] + m2[i][j]) result.append(row) return result def sum_of_elements(matrix): total = 0 for row in matrix: for val in row: total += val return total </pre>

```
def transpose(matrix):
    rows = len(matrix)
    cols = len(matrix[0])
    trans = []
    for j in range(cols):
        row = []
        for i in range(rows):
            row.append(matrix[i][j])
        trans.append(row)
    return trans
print("Enter elements for the Matrix:")
matrix = input_matrix()
while True:
    print("\nMenu Card:")
    print("1. Matrix Addition")
    print("2. Sum of all elements of Matrix")
    print("3. Transpose of Matrix")
    print("4. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        print("Enter elements for another matrix of same order:")
        matrix2 = input_matrix()
        if len(matrix) == len(matrix2) and len(matrix[0]) == len(matrix2[0]):
            result = add_matrices(matrix, matrix2)
            print("Resultant Matrix after Addition:")
            display_matrix(result)
        else:
            print("Error: Matrices must have the same order.")
    elif choice == 2:
        total = sum_of_elements(matrix)
        print("Sum of all elements in the matrix: ", total)
    elif choice == 3:
        trans = transpose(matrix)
        print("Transpose of the Matrix:")
        display_matrix(trans)
    elif choice == 4:
        print("Exiting program...")
        break
    else:
        print("ERROR DUE TO WRONG CHOICE.")
```

Output

```
c:\Users\Himanshu\Downloads\Assignment 6\Assignment 6\Matrix\matrix.py
```

Enter number of rows: 2
Enter number of columns: 2
Enter matrix elements row-wise:
Enter element [1][1]: 1
Enter element [1][2]: 2
Enter element [2][1]: 3
Enter element [2][2]: 4

Menu Card:
1. Matrix Addition
2. Sum of all elements of Matrix
3. Transpose of Matrix
4. Exit
Enter your choice: 3
Transpose of the Matrix:
1 3
2 4

Menu Card:
1. Matrix Addition
2. Sum of all elements of Matrix
3. Transpose of Matrix
4. Exit
Enter your choice: 1
Enter elements for another matrix of same order:
Enter number of rows: 2
Enter number of columns: 2
Enter matrix elements row-wise:
Enter element [1][1]: 4
Enter element [1][2]: 3
Enter element [2][1]: 2
Enter element [2][2]: 1
Resultant Matrix after Addition:
5 5
5 5

Menu Card:
1. Matrix Addition
2. Sum of all elements of Matrix
3. Transpose of Matrix
4. Exit
Enter your choice: 2
Sum of all elements in the matrix: 10

Menu Card:
1. Matrix Addition
2. Sum of all elements of Matrix
3. Transpose of Matrix
4. Exit
Enter your choice: 0

Topic	File Handling
Program 16	<p>Write a menu driven Python program to perform following on contents of a text file. Write separate user defined functions for each option</p> <ol style="list-style-type: none">1. Count number of lines in file.2. Count number of vowels in file3. Count number of words4. Exit <p>Note: Program should raise exception if file doesn't exist</p>

Source
Code

```
def count_lines(filename):
    try:
        file = open(filename, "r")
        lines = file.readlines()
        file.close()
        return len(lines)
    except FileNotFoundError:
        print("FNF.")

def count_vowels(filename):
    try:
        file = open(filename, "r")
        data = file.read()
        file.close()
        count = 0
        vowels = "AEI0Uaeiou"
        for ch in data:
            if ch in vowels:
                count += 1
        return count
    except FileNotFoundError:
        print("Error: File not found.")

def count_words(filename):
    try:
        file = open(filename, "r")
        data = file.read()
        file.close()
        words = data.split()
        return len(words)
    except FileNotFoundError:
        print("Error: File not found.")
        return None
```

```
filename = input("Enter filename plss: ")
while True:
    print("\nMenu Card:")
    print("1. Count number of lines")
    print("2. Count number of vowels")
    print("3. Count number of words")
    print("4. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        result = count_lines(filename)
        if result is not None:
            print("Number of lines in file:", result)
    elif choice == 2:
        result = count_vowels(filename)
        if result is not None:
            print("Number of vowels in file:", result)
    elif choice == 3:
        result = count_words(filename)
        if result is not None:
            print("Number of words in file:", result)
    elif choice == 4:
        print("Exiting program...")
        break
    else:
        print("Critical Error. Try again!!!!")
```

Output

```
tanuj@MacBook-Air Assignment % /  
Enter filename plss: file.txt
```

Menu Card:

1. Count number of lines
2. Count number of vowels
3. Count number of words
4. Exit

Enter your choice: 1

Number of lines in file: 4

Menu Card:

1. Count number of lines
2. Count number of vowels
3. Count number of words
4. Exit

Enter your choice: 2

Number of vowels in file: 6

Menu Card:

1. Count number of lines
2. Count number of vowels
3. Count number of words
4. Exit

Enter your choice: 3

Number of words in file: 5

Menu Card:

1. Count number of lines
2. Count number of vowels
3. Count number of words
4. Exit

Topic	Modules
Program 17	<p>Write a program to create a library in python and import it in a program.</p> <p>Library contains</p> <ul style="list-style-type: none"> Area_rect.py with function rectangle, Area_square.py with function square, Area_Tri.py with function triangle <p>Program do display menu to user and calculate areas accordingly</p>
Source Code	<pre> Area.py > ... 1 from AreaLibrary import Area_rect, Area_square, Area_Tri 2 while True: 3 print("\nRestaurant Menu:") 4 print("1. Area of Rectangle") 5 print("2. Area of Square") 6 print("3. Area of Triangle") 7 print("4. Exit") 8 choice = int(input("Enter your choice: ")) 9 if choice == 1: 10 l = float(input("Enter length of rectangle: ")) 11 b = float(input("Enter breadth of rectangle: ")) 12 area = Area_rect.rectangle(l, b) 13 print("Area of Rectangle is:", area) 14 elif choice == 2: 15 s = float(input("Enter side of square: ")) 16 area = Area_square.square(s) 17 print("Area of Square:", area) 18 elif choice == 3: 19 base = float(input("Enter base of triangle: ")) 20 height = float(input("Enter height of triangle: ")) 21 area = Area_Tri.triangle(base, height) 22 print("Area of Triangle:", area) 23 elif choice == 4: 24 print("Exiting program...") 25 break 26 else: 27 print("Not Working due to wrong choice.") 28 </pre> <pre> AreaLibrary > Area_rect.py > ... 1 def rectangle(length, breadth): 2 area = length * breadth 3 return area 4 </pre>

```
aLibraby > 🐍 Area_square.py > ..
1     def square(side):
2         area = side * side
3         return area
4
aLibraby > 🐍 Area_Tri.py > ...
    def triangle(base, height):
        area = 0.5 * base * height
        return area
```

Output

```
Restaurant Menu:  
1. Area of Rectangle  
2. Area of Square  
3. Area of Triangle  
4. Exit  
Enter your choice: 1  
Enter length of rectangle: 2  
Enter breadth of rectangle: 3  
Area of Rectangle is: 6.0
```

```
Restaurant Menu:  
1. Area of Rectangle  
2. Area of Square  
3. Area of Triangle  
4. Exit  
Enter your choice: 2  
Enter side of square: 12  
Area of Square: 144.0
```

```
Restaurant Menu:  
1. Area of Rectangle  
2. Area of Square  
3. Area of Triangle  
4. Exit  
Enter your choice: 3  
Enter base of triangle: 12  
Enter height of triangle: 32  
Area of Triangle: 192.0
```

```
Restaurant Menu:  
1. Area of Rectangle  
2. Area of Square  
3. Area of Triangle  
4. Exit  
Enter your choice: █
```

Topic	File Handling
Program 18	<p>Write Python code to do following</p> <ol style="list-style-type: none"> 1. Create an Excel file (save as csv) 2. Import file created in part 1 in Python file and display content of csv 3. Append content to file created in part 1 through python. file (read values from user) 4. Display updated csv file from python fil 5.Exit
Source Code	<pre># ''' import csv with open('file.csv', 'r') as h: a = csv.reader(h) for i in a: print(i) with open('file.csv', 'a') as h: values = eval(input("Enter values: ")) b = csv.writer(h) b.writerow(values) with open('file.csv', 'r') as h: a = csv.reader(h) for i in a: print(i) print("Exited") '''</pre>
Output	<pre>tanuj@MacBook-Air Assignment % /usr/local/bin/python3 "/Users/tanuj/Downloads/file.py" ['a', 'd', 'b', 'd', 'd', 'e', 'r', ''] ['a', 'f', 's', 'e', 't', 's', 'r', ''] Enter values: ['a', 'b', 'c', 'd'] ['a', 'd', 'b', 'd', 'd', 'e', 'r', ''] ['a', 'f', 's', 'e', 't', 's', 'r', 'a', 'b', 'c', 'd'] Exited tanuj@MacBook-Air Assignment %</pre>

Topic	File Handling
Program 19	<p>Write a menu driven program to perform following functions on a 2D integer Array with help of user defined functions</p> <p>Menu</p> <ol style="list-style-type: none">1. Sum of elements of particular row number entered by user2. Sum of elements of particular column number entered by user3. Sum of elements at left diagonal4. Sum of elements at right diagonal5. Exit

Source Code

```


def input_matrix():
    rows = 3
    cols = 3
    matrix = []
    print("Enter elements row-wise:")
    for i in range(rows):
        row = []
        for j in range(cols):
            val = int(input("Enter element [" + str(i+1) + "][" + str(j+1) + "]:"))
            row.append(val)
        matrix.append(row)
    return matrix
def display_matrix(matrix):
    print("Matrix is:")
    for row in matrix:
        for val in row:
            print(val, end="\t")
        print()
def sum_of_row(matrix, row_num):
    if row_num < 1 or row_num > len(matrix):
        print("Invalid row number!")
        return
    total = 0
    for val in matrix[row_num - 1]:
        total += val
    print("Sum of elements in row " + str(row_num) + " = " + str(total))
def sum_of_column(matrix, col_num):
    if col_num < 1 or col_num > len(matrix[0]):
        print("Invalid column number!")
        return
    total = 0
    for i in range(len(matrix)):
        total += matrix[i][col_num - 1]
    print("Sum of elements in column " + str(col_num) + " = " + str(total))
def sum_left_diagonal(matrix):
    total = 0
    for i in range(len(matrix)):
        total += matrix[i][i]
    print("Sum of elements on Left Diagonal = " + str(total))
def sum_right_diagonal(matrix):
    total = 0
    n = len(matrix)
    for i in range(n):
        total += matrix[i][n - i - 1]
    print("Sum of elements on Right Diagonal = " + str(total))
matrix = input_matrix()


```

```
print("Sum of elements on Right Diagonal = " + str(total))

matrix = input_matrix()
display_matrix(matrix)
while True:
    print("\nMenu:")
    print("1. Sum of elements of particular row number entered by user")
    print("2. Sum of elements of particular column number entered by user")
    print("3. Sum of elements at left diagonal")
    print("4. Sum of elements at right diagonal")
    print("5. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        r = int(input("Enter row number: "))
        sum_of_row(matrix, r)
    elif choice == 2:
        c = int(input("Enter column number: "))
        sum_of_column(matrix, c)
    elif choice == 3:
        sum_left_diagonal(matrix)
    elif choice == 4:
        sum_right_diagonal(matrix)
    elif choice == 5:
        print("Exiting the program.....")
        break
    else:
        print("Didnt work due to wrong choice")
# ...
```

Output

```

tanuj@MacBook-Air Assignment % /usr/local/bin/python3 "/Users/tanuj/Desktop/Assignment 2.py"
Enter element [1][3]: 3
Enter element [2][1]: 4
Enter element [2][2]: 5
Enter element [2][3]: 6
Enter element [3][1]: 7
Enter element [3][2]: 8
Enter element [3][3]: 9
Matrix is:
1      2      3
4      5      6
7      8      9

Menu:
1. Sum of elements of particular row number entered by user
2. Sum of elements of particular column number entered by user
3. Sum of elements at left diagonal
4. Sum of elements at right diagonal
5. Exit
Enter your choice: 1
Enter row number: 1
Sum of elements in row 1 = 6

Menu:
1. Sum of elements of particular row number entered by user
2. Sum of elements of particular column number entered by user
3. Sum of elements at left diagonal
4. Sum of elements at right diagonal
5. Exit
Enter your choice: 2
Enter column number: 2
Sum of elements in column 2 = 15

Menu:
1. Sum of elements of particular row number entered by user
2. Sum of elements of particular column number entered by user
3. Sum of elements at left diagonal
4. Sum of elements at right diagonal
5. Exit
Enter your choice: 3
Sum of elements on Left Diagonal = 15

Menu:
1. Sum of elements of particular row number entered by user
2. Sum of elements of particular column number entered by user
3. Sum of elements at left diagonal
4. Sum of elements at right diagonal
5. Exit
Enter your choice: 4
Sum of elements on Right Diagonal = 15

Menu:
1. Sum of elements of particular row number entered by user
2. Sum of elements of particular column number entered by user
3. Sum of elements at left diagonal
4. Sum of elements at right diagonal
5. Exit
Enter your choice: 5
Exiting the program.....
tanuj@MacBook-Air Assignment %

```

Topic	File Handling
Program 20	<p>Write a menu driven program to perform following functions on a 1D integer Array with help of user defined functions</p> <p>Menu</p> <ol style="list-style-type: none">1. Insert an element in an ordered Array at user specified position2. Delete an element specified by user (find position then delete)3. Exit

Source Code

```

# 
def input_array():
    n = int(input("Enter number of elements: "))
    arr = []
    print("Enter elements in order:")
    for i in range(n):
        val = int(input("Element " + str(i+1) + ": "))
        arr.append(val)
    return arr
def display_array(arr):
    print("Current Array:", arr)
def insert_element(arr):
    elem = int(input("Enter element to insert: "))
    pos = int(input("Enter position to insert (1-based index): "))
    if pos < 1 or pos > len(arr) + 1:
        print("Invalid position")
        return
    arr.insert(pos - 1, elem)
    print("Element has been inserted successfully.")
def delete_element(arr):
    elem = int(input("Enter element to delete: "))
    if elem in arr:
        arr.remove(elem)
        print("Element has been deleted successfully.")
    else:
        print("Element has not been found in array.")
array = input_array()
display_array(array)
while True:
    print("\nMenu:")
    print("1. Insert an element at a specified position")
    print("2. Delete an element specified by user")
    print("3. Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        insert_element(array)
        display_array(array)
    elif choice == 2:
        delete_element(array)
        display_array(array)
    elif choice == 3:
        print("Exiting the program....")
        break
    else:
        print("No, its wrong choice")
    ...

```

Output

```
tanuj@MacBook-Air Assignment % /usr/local/bin/python
Enter number of elements: 4
Enter elements in order:
Element 1: 4
Element 2: 3
Element 3: 2
Element 4: 1
Current Array: [4, 3, 2, 1]

Menu:
1. Insert an element at a specified position
2. Delete an element specified by user
3. Exit
Enter your choice: 1
Enter element to insert: 2
Enter position to insert (1-based index): 3
Element has been inserted successfully.
Current Array: [4, 3, 2, 2, 1]

Menu:
1. Insert an element at a specified position
2. Delete an element specified by user
3. Exit
Enter your choice: 2
Enter element to delete: 2
Element has been deleted successfully.
Current Array: [4, 3, 1]

Menu:
1. Insert an element at a specified position
2. Delete an element specified by user
3. Exit
Enter your choice: 2
Enter element to delete: 1
Element has been deleted successfully.
Current Array: [4, 3]

Menu:
1. Insert an element at a specified position
2. Delete an element specified by user
3. Exit
Enter your choice: 3
Exiting the program.....
tanuj@MacBook-Air Assignment %
```

Topic	File Handling
Program 1	Student Record mini project
Source Code	<pre> # Created by Tanuj Shah of 12A # mini Project import pickle def file_create(): data={ "roll_nos":[], "names":[], "grade":[], "total_marks": [] } with open("binary.dat", "wb") as fh: pickle.dump(data, fh) return data file_create() def file_read(): try: fh = open("binary.dat", "rb") data = pickle.load(fh) fh.close() return data except FileNotFoundError as errmsg: print(["Error while executing", errmsg]) def file_write(data1): with open("binary.dat", "wb+") as fh: fh.truncate() pickle.dump(data1, fh) return def analyze(data): rec = [] try: for i in range(len(data['roll_nos'])): rec.append([data['roll_nos'][i], data['names'][i], data['grade'][i], data['total_marks'][i]]) except IndexError: rec=[["No Data",]] return rec def countdata(): return len(rec) </pre>

```
def iput(datatype='s', var="value"):
    if datatype == "s":
        try:
            recieived = input(f"Enter the {var} in an integer format: ")
        except:
            recieived=None
    elif datatype == "i":
        try:
            recieived = int(input(f"Enter the {var} in an integer format: "))
        except:
            recieived=None
    return recieived
allow = True
while allow:
    choice = int(input("Enter the choice: "))
    if choice == 1:

        dataset = analyze(file_read())
        print(["Roll Nos", "Name", "Grade", "Total Marks"])
        for i in dataset:
            print(i)
    if choice==2:
        no=input('i', 'roll number')
        name = input("Enter Name: ")
        grade = input("Enter Grade: ")
        mark = input("Enter Total Marks: ")
        reco = file_read()
        reco['roll_nos'].append(no)
        reco['names'].append(name)
        reco['grade'].append(grade)
        reco['total_marks'].append(mark)
        file_write(reco)
    if choice==3:
        datablock = analyze(file_read())
        roll=input('i', 'roll number')
        print(["Roll Nos", "Name", "Grade", "Total Marks"])
        for i in datablock:
            if i[0] == roll:
                print(i)
```

```
if choice==4:
    roll=input('i', 'roll number')
    data = file_read()
    for i in data['roll_nos']:
        if i == roll:
            loc=data['roll_nos'].index(i)
            data['roll_nos'].pop(loc)
            data['names'].pop(loc)
            data[ "grade"].pop(loc)
            data["total_marks"].pop(loc)
            file_write(data)
            print("Operation completed")
if choice==5:
    roll=input('i', 'roll number')
    data = file_read()
    for i in data['roll_nos']:
        if i == roll:
            loc=data['roll_nos'].index(i)
            print("Current Grade:", data[ "grade"][loc])
            change = input("Enter new Grade: ")
            data[ "grade"][loc]=change
            file_write(data)
if choice == 6:
    print("Total number of records: ", len(analyze(file_read())))
if choice == 7:
    allow = False
    print("Exited")
```

Output

```
tanuj@MacBook-Air ~ % /usr/local/bin/python3 /Users/tanuj/Downloads/Python/Assignment/Assignment 1.py
Enter the choice: 1
['Roll Nos', 'Name', 'Grade', 'Total Marks']
Enter the choice: 2
Enter the roll number in an integer format: 1
Enter Name: Tshah
Enter Grade: A
Enter Total Marks: 460
Enter the choice: 3
Enter the roll number in an integer format: 1
['Roll Nos', 'Name', 'Grade', 'Total Marks']
[1, 'Tshah', 'A', '460']
Enter the choice: 4
Enter the roll number in an integer format: 1
Operation completed
Enter the choice: 2
Enter the roll number in an integer format: 1
Enter Name: E
Enter Grade: F
Enter Total Marks: 67
Enter the choice: 5
Enter the roll number in an integer format: 1
Current Grade: F
Enter new Grade: D
Enter the choice: 6
Total number of records: 1
Enter the choice: 7
Exited
tanuj@MacBook-Air ~ %
```

SQL

Topic	SQL																																																																								
Program 1	<p style="text-align: center;">PRACTICE SQL</p> <p>Create a table School with Code number Teachername 20 characters Subject 10 characters DOJ date type hoursperweek number Experience number</p> <p style="text-align: center;">SCHOOL</p> <table border="1"> <thead> <tr> <th>CODE</th><th>TEACHERNAME</th><th>SUBJECT</th><th>DOJ</th><th>HOURS PER WEEK</th><th>EXPERIENCE</th></tr> </thead> <tbody> <tr> <td>1001</td><td>BHUVAN</td><td>COMPUTER</td><td>12/03/2000</td><td>24</td><td>10</td></tr> <tr> <td>1009</td><td>AJAY SINGH</td><td>PHYSICS</td><td>03/09/1998</td><td>26</td><td>12</td></tr> <tr> <td>1203</td><td>REHAAN</td><td>PHYSICS</td><td>09/04/2000</td><td>27</td><td>5</td></tr> <tr> <td>1045</td><td>PHUNSUK WANGDU</td><td>MATHS</td><td>24/08/2000</td><td>24</td><td>15</td></tr> <tr> <td>1123</td><td>SANJAY S</td><td>PHYSICS</td><td>16/07/1999</td><td>28</td><td>3</td></tr> <tr> <td>1167</td><td>RAM SHANKAR</td><td>CHEMISTRY</td><td>19/10/1999</td><td>27</td><td>5</td></tr> <tr> <td>1215</td><td>SANJAY S</td><td>COMPUTER</td><td>11/05/1998</td><td>22</td><td>6</td></tr> </tbody> </table> <p style="text-align: center;">CREATE ANOTHER TABLE ADDRESS</p> <table border="1"> <thead> <tr> <th>CODE</th><th>BLOCK</th><th>LOCATION</th></tr> </thead> <tbody> <tr> <td>1001</td><td>876</td><td>CCK</td></tr> <tr> <td>1009</td><td>456</td><td>KRANJI</td></tr> <tr> <td>1203</td><td>348</td><td>PARIS RIS</td></tr> <tr> <td>1045</td><td>765</td><td>DOVER</td></tr> <tr> <td>1123</td><td>986</td><td>BUGIS</td></tr> <tr> <td>1167</td><td>234</td><td>WOODLANDS</td></tr> <tr> <td>1215</td><td>SANJAY S</td><td>COMPUTER</td></tr> </tbody> </table>	CODE	TEACHERNAME	SUBJECT	DOJ	HOURS PER WEEK	EXPERIENCE	1001	BHUVAN	COMPUTER	12/03/2000	24	10	1009	AJAY SINGH	PHYSICS	03/09/1998	26	12	1203	REHAAN	PHYSICS	09/04/2000	27	5	1045	PHUNSUK WANGDU	MATHS	24/08/2000	24	15	1123	SANJAY S	PHYSICS	16/07/1999	28	3	1167	RAM SHANKAR	CHEMISTRY	19/10/1999	27	5	1215	SANJAY S	COMPUTER	11/05/1998	22	6	CODE	BLOCK	LOCATION	1001	876	CCK	1009	456	KRANJI	1203	348	PARIS RIS	1045	765	DOVER	1123	986	BUGIS	1167	234	WOODLANDS	1215	SANJAY S	COMPUTER
CODE	TEACHERNAME	SUBJECT	DOJ	HOURS PER WEEK	EXPERIENCE																																																																				
1001	BHUVAN	COMPUTER	12/03/2000	24	10																																																																				
1009	AJAY SINGH	PHYSICS	03/09/1998	26	12																																																																				
1203	REHAAN	PHYSICS	09/04/2000	27	5																																																																				
1045	PHUNSUK WANGDU	MATHS	24/08/2000	24	15																																																																				
1123	SANJAY S	PHYSICS	16/07/1999	28	3																																																																				
1167	RAM SHANKAR	CHEMISTRY	19/10/1999	27	5																																																																				
1215	SANJAY S	COMPUTER	11/05/1998	22	6																																																																				
CODE	BLOCK	LOCATION																																																																							
1001	876	CCK																																																																							
1009	456	KRANJI																																																																							
1203	348	PARIS RIS																																																																							
1045	765	DOVER																																																																							
1123	986	BUGIS																																																																							
1167	234	WOODLANDS																																																																							
1215	SANJAY S	COMPUTER																																																																							

	<p>Write SQL commands for the following :-</p> <ol style="list-style-type: none">1. Show all records.2. Display the teachername and experience of all.3. List the teachername and Date of joining of those whose subject is English.4. Show the names and subject of those whose experience in less than 10 years.5. Show the names and hours of those whose experience is between 8 to 15 years.6. Show the code of those who teach MATHS or COMPUTER.7. Show the code of those whose subject is neither English, physics nor chemistry using IN operator8. Show the teachers who have joined in the year 1998.9. Display the names of those who have joined before end of dec – 2000.10. List the names of those whose subject is not Maths. 12. Show the subject taught by BHUVAN. <code>SELECT SUBJECT FROM SCHOOL WHERE TEACHERNAME="BHUVAN"</code>
--	---

	<p>11. List the subjects which are not taught by REHAAN, SANJAY S AND AJAY SINGH</p> <p>12. Add a new column HOURLY_WAGE NUMBER(6,2) and TOTAL_WAGE NUMBER(6,2).</p> <p>13. Assign the HOURLY_WAGE of 100 to those whose Experience is more than 10 years.</p> <ul style="list-style-type: none"> ➤ UPDATE SCHOOL SET H_WAGE=100 WHERE EXPERIENCE > 10; <p>14. Assign the HOURLY_WAGE of 75 to those whose Experience is more than between 5 to 10.</p> <p>15. Assign the HOURLY_WAGE of 100 to those whose Experience is less than 5 years.</p> <p>16. Find out the TOTAL_WAGE for all the teachers as HOURSPERWEEK * HOURLY_WAGE</p> <p>17. Insert the following rows in the table :- 1017, 'MANGAL' NULL '10/03/2000' 20, 15, 60 1076 'RAJ' NULL, '20/12/2009', 30, 7, 80</p> <p>18. Show all the subjects taught in the school. ➤ SELECT DISTINCT SUBJECT FROM SCHOOL;</p>
--	---

19. Sort the table in the increasing order of experiences showing all the columns.

20. Sort the table showing name and subject in the decreasing order of date of joining.

21. Sort the table showing name and TOTAL_WAGE in the decreasing order of hours per week who have taught for less than 26 hours.

22. Show the teachers whose name starts with 'H'.

23. Show the teachers whose name ends with 'N'.

24. Show the teachers whose name contains letter R.

25. Show the teachers whose name is 5 letters long.

26. Show the teachers whose name has atleast 5 letters.

27. Show the names of teachers where subject is not yet allotted.

28. Find the sum of experiences for physics teachers.

	<p>29. Find the maximum experience in the table.</p> <p>30. Find out how many teachers have experience of more than 8 years.</p> <p>31. Find out the sum of experiences of teachers for all the subjects.</p> <p>32. Find out the average of TOTAL_WAGE for all the subjects whose experience is more than 8 years.</p> <p>33. Find out how many teachers are there for each subject only if there are more than 2 teachers.</p> <p>34. Find the average experience of those whose subject is 'COMPUTER'</p> <p>35. Change the experience of SANJAY to 5 years.</p> <p>36. Increase the experience of all maths teachers by 3 years.</p> <p>37. Change the size of teachername column to 25 characters.</p> <p>38. Allot subject "COMPUTERS" to those who are not yet allotted the subject.</p>
--	---

	<p>39. Change the code of REHAAN to 1214 and hourperweek to 24.</p> <p>40. Create a view T_SUB which have the teachers name and Subject of all.</p> <p>41. Create a view T_EXP which consists of teachers name and experience whose experience is more than 5 years.</p> <p>42. Delete the above two views.</p> <p>43. Use the SET operator UNION to find out the teachers name for the subjects COMPUTER AND PHYSICS.</p> <p>44. Use the SET operator INTERSECT to find out the teachers name for the subjects COMPUTER AND PHYSICS.</p> <p>45. Use the SET operator INTERSECT to find out the teachers name for the subjects COMPUTER AND PHYSICS</p> <p>46. Show the names of the teachers, subject and the location they stay in.</p> <p>47. Show the names of the teachers, subject and the location they stay only for those whose experience is more than 10 years.</p>
--	--

	<p>48. Remove the record of those who have joined before mid of 2001.</p> <p>49. Remove all the records whose experience is less than 5 years</p> <p>50. Remove all records.</p> <p>51. Remove entire table.</p>
Source Code	<p>Initialize Tables</p> <pre>CREATE TABLE School(Code INT, TeacherName Char(20), Subject Char(10), DOJ Date, HoursPerWeek INT, Experience INT);</pre> <p>INSERT INTO School Values (1001,'BHUVAN','COMPUTER','2000-03-12',24,10);</p> <p>INSERT INTO School Values (1009,'AJAY SINGH','PHYSICS','1998-09-03',26,12);</p> <p>INSERT INTO School Values (1203,'REHAAN','PHYSICS','2000-04-09',27,5);</p> <p>INSERT INTO School Values (1045,'PHUNSUK WANGDU ','MATHS','2000-08-24',24,15);</p> <p>INSERT INTO School Values (1123,'SANJAY S','PHYSICS','1999-07-16',28,3);</p>

	<pre>INSERT INTO School Values (1167,'RAM SHANKAR','CHEMISTRY','1999-10-19',27,5); INSERT INTO School Values (1215,'SANJAY S','COMPUTER','1998-05-11',22,6); CREATE TABLE Address(Code INT, Block VARCHAR, Location Char(30)); INSERT INTO Address VALUES (1001,'876','CCK'); INSERT INTO Address VALUES (1009,'456','KRANJI'); INSERT INTO Address VALUES (1203,'348','PARIS RIS'); INSERT INTO Address VALUES (1045,'765','DOVER'); INSERT INTO Address VALUES (1123,'986','BUGIS'); INSERT INTO Address VALUES (1167,'234','WOODLANDS'); INSERT INTO Address VALUES (1215,'SANJAY S','COMPUTER'); Q1) SELECT * FROM Address; SELECT * FROM School;</pre>
--	--

Output

Output					
Code	Block		Location		
1001	876		CCK		
1009	466		KRANJI		
1203	348		PARIS RS		
1045	765		DOVER		
1123	986		BUGIS		
1167	234	SANJAY S	WOODLANDS		
1215			COMPUTER		

Code	TeacherName	Subject	DOJ	HoursPerWeek	Experience
1001	BHUVAN	COMPUTER	2000-03-12	24	10
1009	AJAY SINGH	PHYSICS	1998-09-03	26	12
1203	REHAAN	PHYSICS	2000-04-09	27	5
1045	PHUNSUK WANGDU	MATHS	2000-08-24	24	15
1123	SANJAY S	PHYSICS	1999-07-16	28	3
1167	RAM SHANKAR	CHEMISTRY	1999-10-19	27	5
1215	SANJAY S	COMPUTER	1998-05-11	22	6

Q2)

SELECT TeacherName, Experience FROM School;

Output	
TeacherName	Experience
BHUVAN	10
AJAY SINGH	12
REHAAN	5
PHUNSUK WANGDU	15
SANJAY S	3
RAM SHANKAR	5
SANJAY S	6

Q3)

SELECT Teachername, DOJ FROM School WHERE Subject=='English';

Output	
SQL query successfully executed. However, the result set is empty.	

Q4)

SELECT Teachername, Subject FROM School WHERE Experience < 10;

Output

TeacherName	Subject
REHAAN	PHYSICS
SANJAYS	PHYSICS
RAM SHANKAR	CHEMISTRY
SANJAYS	COMPUTER

Q5)

SELECT Teachername, HoursPerWeek FROM School WHERE Experience BETWEEN 8 AND 15;

TeacherName	HoursPerWeek
BHUVAN	24
AJAY SINGH	26
PHUNSUK WANGDU	24

Q6)

SELECT Code FROM School WHERE Subject = 'MATHS' OR Subject = COMPUTER;

Output

Code
1001
1045
1215

Q7)

SELECT Code FROM School WHERE NOT Subject IN ('ENGLISH', 'PHYSICS', 'CHEMISTRY');

Output

Code
1001
1045
1215

Q8)

```
SELECT TeacherName FROM School WHERE strftime('%Y', DOJ) = '1998';
```

Output

TeacherName
AJAY SINGH
SANJAY S

Q9)

```
SELECT TeacherName FROM School WHERE DOJ < '2000-12-31';
```

Output

TeacherName
BHUVAN
AJAY SINGH
REHAAN
PHUNSUK WANGDU
SANJAY S
RAM SHANKAR
SANJAY S

Q10)

```
SELECT TeacherName FROM School WHERE NOT Subject='MATHS';
```

Output

```
TeacherName
BHUVAN
AJAY SINGH
REHAAN
SANJAY S
RAM SHANKAR
SANJAY S
```

Q11)

-

Q12)

```
SELECT SUBJECT FROM School WHERE TeacherName='BHUVAN';
```

Output

```
Subject
COMPUTER
```

Q11)

```
SELECT SUBJECTS FROM School WHERE SUBJECT NOT IN ("REHAAN", "SANJAY S", "AJAY SINGH")
```

Q12)

```
ALTER TABLE School
```

```
ADD HOURLY_WAGE NUMBER(6,2);
```

```
ALTER TABLE School
```

```
ADD TOTAL_WAGE
```

```
NUMBER(6,2);
```

Q13)

UPDATE SCHOOL

SET H_WAGE=100

	<pre> WHERE EXPERIENCE > 10; Q14) UPDATE SCHOOL SET H_WAGE=75 WHERE EXPERIENCE BETWEEN 5 AND 10; Q15) UPDATE SCHOOL SET H_WAGE=100 WHERE EXPERIENCE < 5; Q16) SELECT HOURSPERWEEK * HOURLY_WAGE AS TOTAL_WAGE FROM School; Q17) INSERT INTO School VALUES (1017, 'MANGAL' NULL '10/03/2000' 20, 15, 60), (1076 'RAJ' NULL, '20/12/2009', 30, 7, 80) Q18) SELECT DISTINCT SUBJECT FROM SCHOOL; Q19) SELECT * FROM School ORDER BY Experiences; Q20) SELECT NAME, SUBJECT FROM School ORDER BY Date_of_joining DESC; Q21) SELECT Name, TOTAL_WAGE FROM School WHERE Hours_per_week < 26 ORDER BY Hours_Per_Week DESC; Q22) SELECT DISTINCT NAME FROM School WHERE Name LIKE 'H%'; Q23) SELECT DISTINCT NAME FROM School WHERE Name LIKE '%N'; </pre>
--	--

	<p>Q24)</p> <pre>SELECT DISTINCT NAME FROM School WHERE Name LIKE '%R%';</pre> <p>Q25)</p> <pre>SELECT DISTINCT NAME FROM School WHERE Name LIKE '_____';</pre> <p>Q26)</p> <pre>SELECT DISTINCT NAME FROM School WHERE Name LIKE '_____ %';</pre> <p>Q27)</p> <pre>SELECT DISTINCT NAME FROM School WHERE Subject IS NULL;</pre> <p>Q28)</p> <pre>SELECT Sum(Experiences) FROM School WHERE Subject ='Physics';</pre> <p>Q29)</p> <pre>SELECT Max(Experiences) FROM School;</pre> <p>Q30)</p> <pre>SELECT Name FROM School WHERE Experience > 8;</pre> <p>Q31)</p> <pre>SELECT Sum(Experiences) FROM School;</pre> <p>Q32)</p> <pre>SELECT AVERAGE(Total_Wage) FROM School WHERE Experience > 8;</pre> <p>Q33)</p> <pre>SELECT Subject, COUNT(TEACHERS) FROM School GROUP BY TEACHERS IF COUNT(TEACHERS)>2;</pre> <p>Q34)</p> <pre>SELECT AVERAGE(EXPERIENCE) FROM School WHERE SUBJECT ='Computer';</pre> <p>Q35)</p>
--	--

	<pre>UPDATE School SET Experience = 5 WHERE NAME='Sanjay'; Q36) UPDATE School SET Experience = Experience+3 WHERE Subject='Math'; Q37) ALTER TABLE School MODIFY TEACHERNAME CHAR(25); Q38) UPDATE School SET Subject ='Computer' WHERE Subject IS NULL; Q39) UPDATE School SET Code=1214, Hourperweek=25 WHERE Name='Rehaan'; Q40) CREATE VIEW T_SUB SELECT Teachers_Name, Subject FROM School; Q41) CREATE VIEW T_EXP SELECT Teachers_Name, Experience FROM School WHERE Experience>5; Q42) DROP VIEW T_EXP, T_SUB; Q43)</pre>
--	--

SELECT Name FROM School WHERE SUBJECT="Computer' UNION SELECT Name FROM School WHERE Subject='Physics';

Q44)

SELECT Name FROM School WHERE SUBJECT="Computer' INTERSECT SELECT Name FROM School WHERE Subject='Physics';

Q45)

SELECT Name FROM School WHERE SUBJECT="Computer' INTERSECT SELECT Name FROM School WHERE Subject='Physics';

Q46)

SELECT Names, Subject, Address FROM School;

Q48)

DELETE * FROM School WHERE DOJ<01-07-2001;

Q49)

DELETE * FROM School WHERE Experience<5;

Q50)

DROP TABLE School;



Topic	SQL Connector
Program 1	<p>1 State the purpose of the following Python MySQL functions: a) connect() b) commit()</p>
	<p>2 Differentiate between the following cursor functions: a) fetchone() and fetchall() b) execute() and executemany()</p>
	<p>3 Explain the use of the %s format specifier while writing SQL queries in Python. Give a short code example demonstrating its use in an INSERT query.</p>
	<p>4 Write a Python program to connect with a MySQL database named COLLEGE and perform the following:</p> <ul style="list-style-type: none"> • Create a cursor object. • Increase the price of all books in the table Library by 10%. • Display the number of records updated using rowcount. • Display all records from the table after the update. <p>Use the functions: connect(), cursor(), execute(), commit(), fetchall().</p>
	<p>5 Write a Python program to connect to a database SCHOOL, insert a new record into a table Student (fields: admno, name, marks), and then display all records from the table. Use the %s format specifier for the query.</p>
	<p>6 Meera wants to write a program in Python to do the following in the table named Employee in the MySQL database OFFICE:</p> <ul style="list-style-type: none"> • emp_id (Employee ID) — integer • emp_name (Employee Name) — string • dept (Department) — string • salary — float <p>Note the following details to establish connectivity between Python and MySQL:</p> <ul style="list-style-type: none"> • Username – root • Password – 12345678 • Host – localhost <p>i) The values of fields emp_id, emp_name, dept, and salary must be accepted from the user and inserted into the table Employee. ii) Update the salary of all employees in a user-specified department by 5%.</p> <p>Help Meera to complete the program in Python.</p>

Source Code

```

# Tanuj Shah 12 A CS Test 7th November 2025
# Q1)
    a) connect() is used to establish connection between the python script and the sql server with arguments such as host, user, passwd, database
    b) commit() Is used to write changes into the SQL database
    ''
    ''
    a) fetchone() fetches only one row (the topmost) of the data returned by select statement whereas fetchall() fetches all the returned data into a variable
    b) execute() is used to execute a single SQL statement whereas executemany() can be used to execute multiple SQL statements at the same time.
    ''
    ''
    # Q3)
    ''
    %s is part of the old style of formatting used to fill values into a string directly through variables
Example:
table_name = "Students"
values=(10,20,30)
command = "INSERT INTO %s VALUES %s;"%(table_name, values)
    ''
    # Q4)
import mysql.connector as m
con = m.connect(host="localhost",user="root",passwd="12345678",database="COLLEGE")
cur = con.cursor()
statement1 = "UPDATE Library SET price=price+(0.10*price);"
cur.execute(statement1)
con.commit()
statement2="SELECT ROWCOUNT(price) FROM Library;"
cur.execute(statement2)
data = cur.fetchall()
print(data)
statement3 = "SELECT * FROM Library"
cur.execute(statement3)
data2=cur.fetchall()
print(data2)
    #
# Q5)
import mysql.connector as m
con = m.connect(host="localhost",user="root",passwd="12345678",database="SCHOOL")
cur = con.cursor()
admno=0
name="test"
marks=0
statement1="INSERT INTO Student VALUES %s , %s , %s; "%(admno,name,marks)
cur.execute(statement1)
con.commit()
statement2="SELECT * FROM Students;"
cur.execute(statement2)
data = cur.fetchall()
print(data)
    #
# Q6)
import mysql.connector as m
con = m.connect(host="localhost",user="root",passwd="12345678",database="OFFICE")
cur = con.cursor()
emp_id = int(input("Enter Employee ID: "))
emp_name=str(input("Enter Employee Name: "))
dept = str(input("Enter dept: "))
salary=float(input("Enter salary"))
statement1= "INSERT INTO Employee Values %s, %s, %s, %s;"%(emp_id,emp_name,dept,salary)
cur.execute(statement1)
con.commit()
dept = str(input("Enter dept: "))
statement2 = "UPDATE Employee SET salary=salary+salary%0.05 WHERE dept=%s;"%(dept,)
cur.execute(statement2)
con.commit()

```