OBJECT ORIENTED PROGRAMMING USING PYTHON
LAB FILE

Manav Rachna International Institute of Research and Studies

School of Computer Applications

Department of Computer Applications

Submitted By	
Student Name	Abhinav Mishra
Roll No	24/SCA/BCA(AI&ML)/002
Programme	BCA (AI&ML)
Semester	3
Section/Group	C
Department	School of Computer Applications
Session / Batch	2024-27
Submitted To	
Faculty Name	Dr Sakshi Gupta

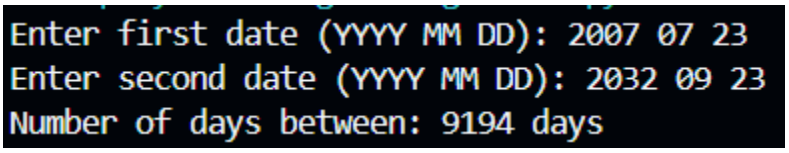
MANAV RACHNA
vidyaparikshal
Manav Rachna International Institute of Research and Studies
Deemed to be University under section 3 of the UGC Act, 1956

SCHOOL OF COMPUTER APPLICATIONS

1. Calculate number of days between two dates

Code - from datetime import date

```
# Take input from user
y1, m1, d1 = map(int, input("Enter first date (YYYY MM DD): ").split())
y2, m2, d2 = map(int, input("Enter second date (YYYY MM DD): ").split())

# Create date objects
date1 = date(y1, m1, d1)
date2 = date(y2, m2, d2)
print(f"Number of days between: {delta.days} days")

# Difference
delta = abs(date2 - date1)
```



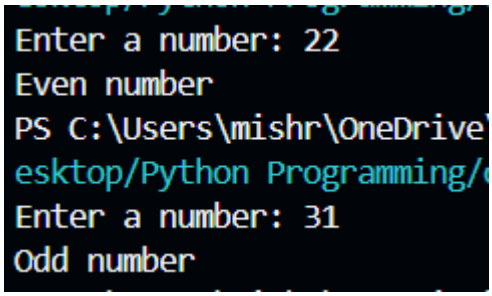
2. Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn

Code - n = int(input("Enter an integer: "))
result = n + int(str(n)*2) + int(str(n)*3)
print("Result:", result)

```
Enter an integer: 23
Result: 234669
```

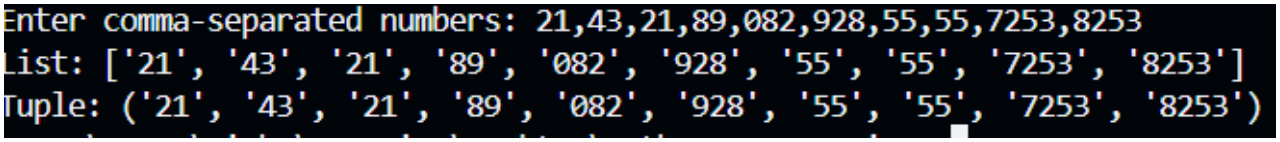
3. Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. Hint: how does an even / odd number react differently when divided by 2?
Code - num = int(input("Enter a number: "))

```
if num % 2 == 0:
    print("Even number")
else:
    print("Odd number")
```



4. Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.

```
values = input("Enter comma-separated numbers: ")

# Split into list
list_values = values.split(",")

# Convert to tuple
tuple_values = tuple(list_values)

print("List:", list_values)
print("Tuple:", tuple_values)
```



5. Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum.

Code -

```
def sum_three(a, b, c):
    total = a + b + c
    if a == b == c:
        return total * 3
    return total

# Taking input from user
a, b, c = map(int, input("Enter three numbers separated by space: ").split())

# Calling function and printing result
print("Result:", sum_three(a, b, c))
```

```
Enter three numbers separated by space: 4 5 6
Result: 15
```

6. Write a Python program to test whether a passed letter is a vowel or not
Code- letter = input("Enter a letter: ").lower()

```
if letter in 'aeiou':
    print(f"{letter} is a vowel")
else:
    print(f"{letter} is not a vowel")
```

Output-

Q7- Take a list, say for example this one:
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
and write a program that prints out all the elements of the list that are less than 5.
Extras:Program-

a. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.

b.Write this in one line of Python.
c.Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.
Ans

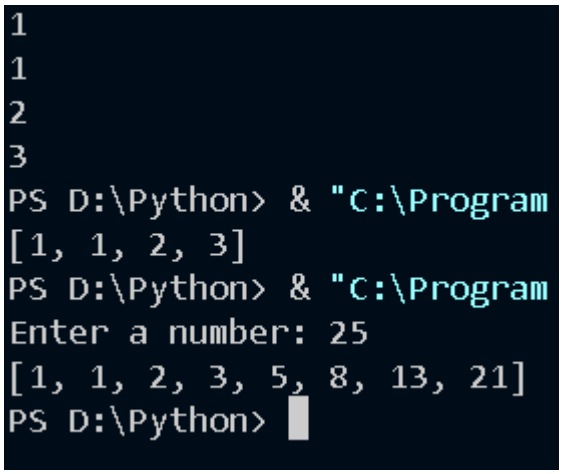
```
a = [1,1,2,3,5,8,13,21,34,55,89]
for i in a:
   if i < 5:
      print(i)
```

b-

```
print([i for i in a if i<5])
```

c-

```
a = [1,1,2,3,5,8,13,21,34,55,89]
num = int(input("Enter a number: "))
result = [x for x in a if x < num]
print(result)
```



Q8- Create a program that asks the user for a number and then prints out a list of all the divisors of that number.

Ans-
```
num = int(input("Enter a number: "))
divisors = []

for i in range(1, num + 1):
    if num % i == 0:
        divisors.append(i)

print("Divisors of", num, "are:", divisors)
```

```
Divisors of 800 are: [1, 2, 4, 5, 8, 10, 16, 20, 25, 32, 40, 50, 80, 100, 160, 200, 400, 800]
```

Q9- Take two lists, say for example these two:
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of
different sizes.

Program-

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
common = list(set(a) & set(b))
print("Common elements:", common)
```

OUTPUT-

Q10- Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)
Ans-

```
string = input("Enter a string: ")
string = string.lower()
string = string.replace(" ", "")
if string == string[::-1]:
    print("The string is a palindrome")
else:
    print("The string is not a palindrome")
```

Output-

```
The string is a palindrome
```

Q11- Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.
Ans-

```
a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
even_elements = [element for element in a if element % 2 == 0]
print("Even Elements are:",even_elements)
```

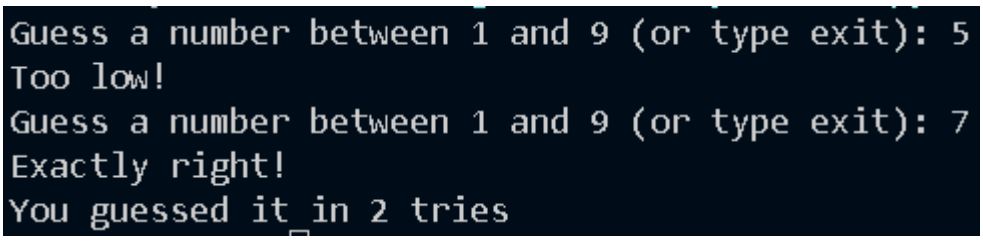
Output-

```
Even Elements are: [4, 16, 36, 64, 100]
```

Q12- Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. (Hint: remember to use the user input lessons from thevery first exercise)
Ans-

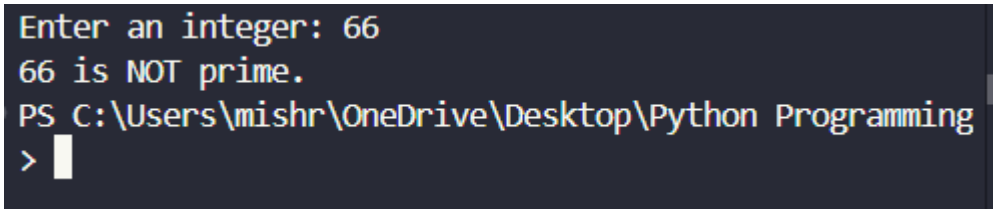
```
import random

number = random.randint(1, 9)
guess = 0
count = 0

while guess != number and guess != "exit":
    guess = input("Guess a number between 1 and 9 (or type exit): ")

    if guess == "exit":
        break

    guess = int(guess)
    count += 1

    if guess < number:
        print("Too low!")
    elif guess > number:
        print("Too high!")
    else:
        print("Exactly right!")
        print("You guessed it in", count, "tries")
```

Output-



13.Ask the user for a number and determine whether the number is prime or not.

code-
```
def is_prime(n: int) -> bool:
    if n <= 1:
        return False
    if n <= 3:
        return True
    if n % 2 == 0:
        return False
    i = 3
    while i * i <= n:
        if n % i == 0:
            return False
        i += 2
    return True

num = int(input("Enter an integer: ").strip())
print(f"{num} is prime." if is_prime(num) else f"{num} is NOT prime.")
```



Q14- Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

Ans-
```
def remove_duplicates(input_list):
    seen = set()
    result = []
    for item in input_list:
        if item not in seen:
            result.append(item)
            seen.add(item)
    return result
a = [1, 4, 9, 16, 25, 25, 49, 4, 100, 36, 49, 64, 81, 100]
print(remove_duplicates(a))
```

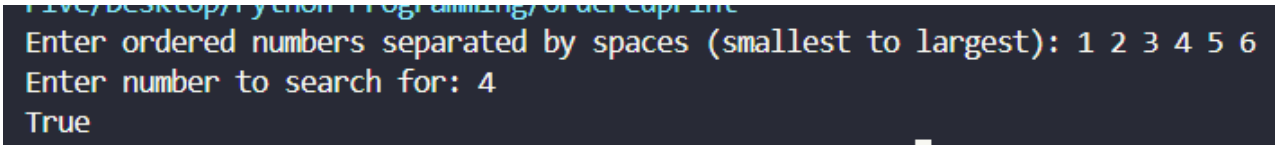
Output-

```
[1, 4, 9, 16, 25, 49, 100, 36, 64, 81]
```

Q15- Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether
or not the given number is inside the list and returns (then prints) an appropriate boolean.

Ans-
```
def is_number_in_list(ordered_list, number):
    left, right = 0, len(ordered_list) - 1
    while left <= right:
        mid = (left + right) // 2
        if ordered_list[mid] == number:
            return True
        elif ordered_list[mid] < number:
            left = mid + 1
        else:
            right = mid - 1
    return False
my_list = [1, 3, 5, 7, 9, 11, 13, 15]
num_to_check = 7
result = is_number_in_list(my_list, num_to_check)
print("Result of first list ")
print(result)

num_to_check = 8
result = is_number_in_list(my_list, num_to_check)
print("Result of second list result")
print(result)
```

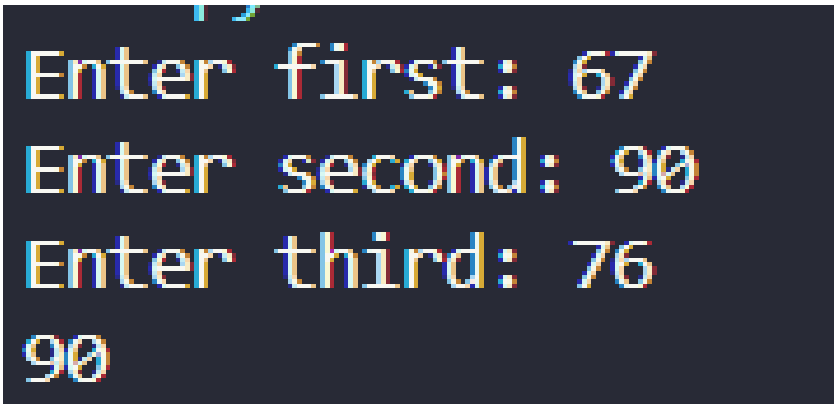
Output-



16. Implement a function that takes three variables and returns the largest without using max().

ANSWER:

```
def largest_of_three(a, b, c):
    largest = a
    if b > largest:
        largest = b
    if c > largest:
        largest = c
    return largest

a = int(input("Enter first: "))
b = int(input("Enter second: "))
c = int(input("Enter third: "))
print(largest_of_three(a, b, c)
```

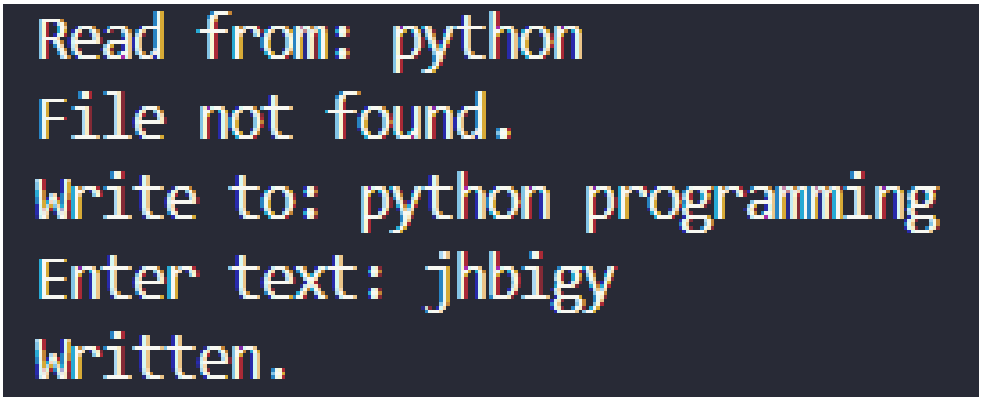


17. Python program to perform read and write operations on a file.

ANSWER:

```
src = input("Read from: ")
try:
    with open(src, "r") as f:
        content = f.read()
    print(content)
except:
    print("File not found.")

dst = input("Write to: ")
text = input("Enter text: ")
with open(dst, "w") as f:
    f.write(text)
print("Written.")
```

OUTPUT:

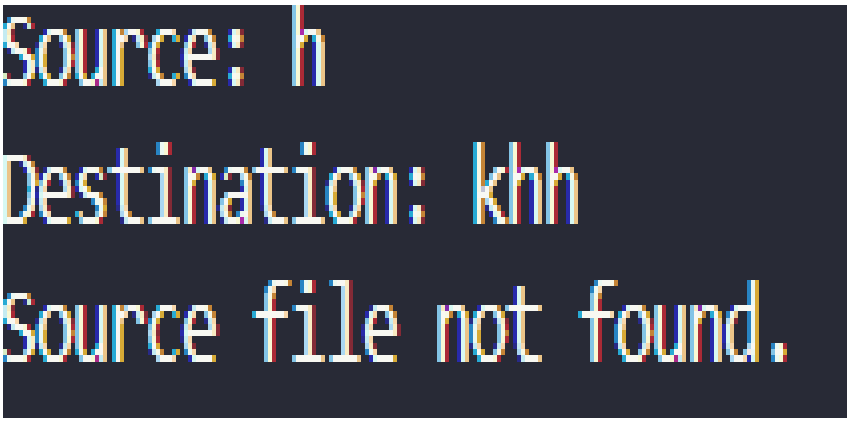


18. Python program to copy the contents of a file to another file.

ANSWER:

```
src = input("Source: ")
dst = input("Destination: ")
try:
    with open(src, "rb") as f1, open(dst, "wb") as f2:
        f2.write(f1.read())
    print("Copy done.")
except:
    print("Source file not found.")
```

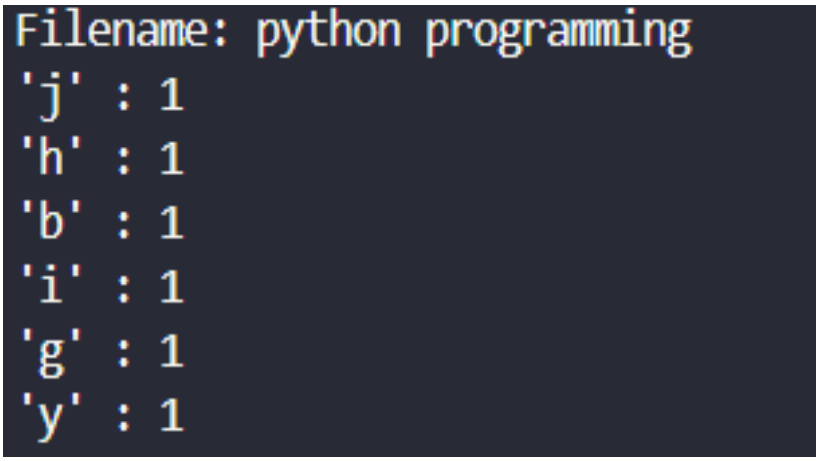
OUTPUT:



19. Python program to count frequency of characters in a given file.
ANSWER:

```
from collections import Counter

fname = input("Filename: ")
try:
    with open(fname, "r") as f:
        text = f.read()
    freq = Counter(text)
    for c, ct in freq.items():
        print(repr(c), ":", ct)
except:
    print("File not found.")
```

OUTPUT:



20. Python program to print each line of a file in reverse order.
ANSWER:

```
fname = input("Filename: ")
try:
    with open(fname, "r") as f:
        for line in f:
            print(line.rstrip()[::-1])
except:
    print("File not found.")
```

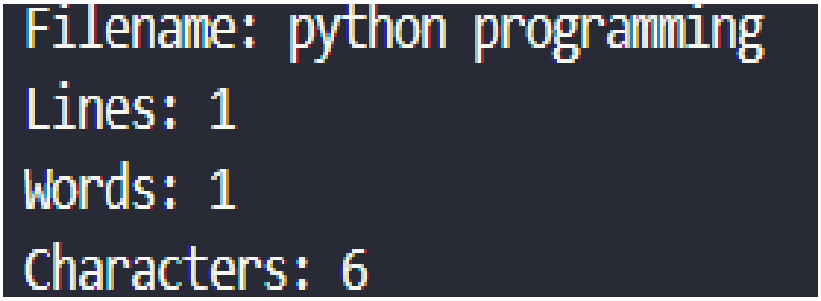
OUTPUT

21. Python program to compute the number of characters, words and lines in a file.

ANSWER:

```
fname = input("Filename: ")
try:
    with open(fname, "r") as f:
        lines = f.readlines()
    chars = sum(len(l) for l in lines)
    words = sum(len(l.split()) for l in lines)
    print("Lines:", len(lines))
    print("Words:", words)
    print("Characters:", chars)
except:
    print("File not found.")
```

OUTPUT:

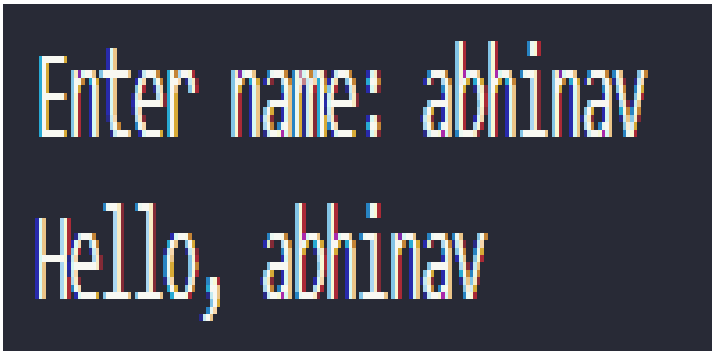


22. Program that greets the user unless their name is Rahul, then throws an exception.

ANSWER:

```
name = input("Enter name: ")
if name.lower() == "rahul":
    raise Exception("Rahul not allowed.")
else:
    print("Hello,", name)
```

OUTPUT:

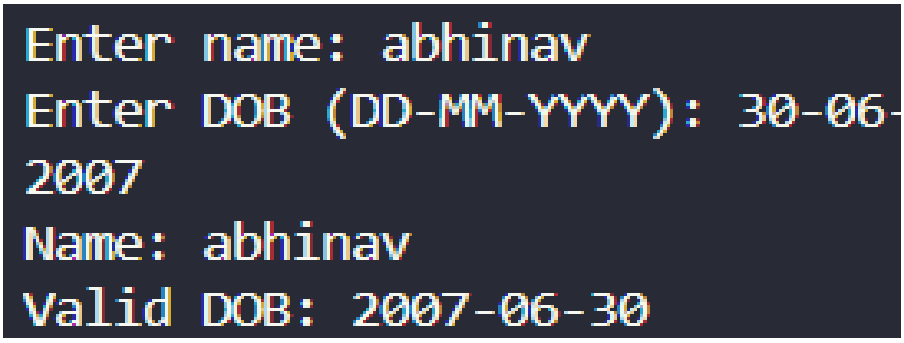


23. Program that accepts personal details and DOB, raising exception for invalid date.

ANSWER:

```
from datetime import datetime

name = input("Enter name: ")
dob = input("Enter DOB (DD-MM-YYYY): ")

try:
    d = datetime.strptime(dob, "%d-%m-%Y")
    print("Name:", name)
    print("Valid DOB:", d.date())
except:
    raise Exception("Invalid date entered.")
```

OUTPUT:

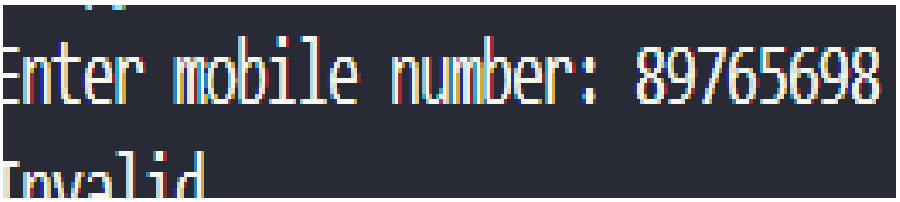


24. Regular Expression to check valid 10-digit mobile number starting with 7,8 or 9.

ANSWER:

```
import re

pattern = r'^[789]\d{9}$'
num = input("Enter mobile number: ")

if re.match(pattern, num):
    print("Valid")
else:
    print("Invalid")
```

OUTPUT:



25. Program that checks spelling mistakes using a dictionary file.
ANSWER:

```
import string

dict_file = input("Dictionary file: ")
text_file = input("Text file: ")

try:
    with open(dict_file) as f:
        words = {w.strip().lower() for w in f}
except:
    print("Dictionary not found.")
    words = set()

try:
    with open(text_file) as f:
        text = f.read()
except:
    print("Text file not found.")
    text = ""

clean = []
for w in text.split():
    w = w.strip(string.punctuation).lower()
    if w:
        clean.append(w)

miss = sorted([w for w in clean if w not in words])

if miss:
    print("Misspelled:")
    for w in miss:
        print(w)
else:
    print("No mistakes.")
```

OUTPUT:

```
Dictionary file: PYTHON
Text file: NON
Dictionary not found.
Text file not found.
No mistakes.
```