

Musaliar College of Engineering & Technology

MUSALIAR COLLEGE P.O., PATHANAMTHITTA - 689 653



LABORATORY RECORD

Certified that this is the Bonafide Record of the work done by

Sri/Smt.....of.....Semester

Class of (Reg No.....)

of.....Branch

in the.....Laboratory

during the academic year 20 - 20

Name of Examination:

Internal Examiner

External Examiner

Staff in-charge

DEPARTMENT OF COMPUTER APPLICATIONS

VISION

“To produce competent and dynamic professionals in the field of Computer Applications to thrive and cater the changing needs of the society through research and education”.

MISSION

To impart high quality technical education and knowledge in Computer Applications. To introduce moral, ethical and social values to Computer Application students. To establish industry institute interaction to enhance the skills of Computer Application students. To promote research aimed towards betterment of society.

INDEX

SL.NO	DATE	NAME OF EXPERIMENT	PAGE NO.	REMARKS
1	12/09/24	FUTURE LEAP YEAR FROM CURRENT YEAR TO FINAL YEAR	1-2	
2	12/09/24	GENERATE POSITIVE LIST OF NUMBERS FROM A LIST	3-4	
3	12/09/24	SQUARE OF N NUMBERS	5-6	
4	4/10/24	FORM A LIST OF VOWELS OF A WORD	7-8	
5	4/10/24	LIST ORDINAL VALUE OF EACH ELEMENT OF A WORD	9-10	
6	4/09/24	OCCURRENCE OF EACH WORD	11-12	
7	4/09/24	LIST OF INTEGERS	13-14	
8	4/09/24	COUNT THE OCCURRENCES	15-16	
9	4/09/24	LIST COMPARISON	17-18	
10	10/10/24	OCCURRENCE OF FIRST CHARACTER REPLACED	19-20	
11	10/10/24	FIRST AND LAST CHARACTER EXCHANGE	21	
12	18/10/24	AREA OF CIRCLE	22	

13	18/10/24	BIGGEST OF 3 NUMBERS	23-24	
14	18/10/24	TO PRINT FILENAME EXTENSION	25	
15	18/10/24	DISPLAY FIRST AND LAST COLORS FROM LIST OF COLORS	26	
16	18/10/24	INTEGER OPERATIONS	27	
17	18/10/24	PRINT COLORS IN COLOR-LIST1 NOT CONTAINED IN COLOR-LIST2	28	
18	24/10/24	GENERATE SINGLE STRING FROM TWO STRINGS BY SWAPPING CHARACTER AT POSITION 1	29	
19	24/10/24	SORT DICTIONARY IN ASCENDING AND DESCENDING	30	
20	25/10/24	MERGE 2 DICTIONARIES	31	
21	25/10/24	GCD OF 2 NUMBERS	32	
22	25/10/24	REMOVE EVEN NUMBERS FROM A LIST	33-34	
23	1/11/24	FACTORIAL OF A NUMBER	35-36	
24	1/11/24	FIBONACCI SERIES	37-38	
25	1/11/24	SUM OF ALL ITEMS IN A LIST	39-40	
26	1/11/24	PRINT THE PERFECT SQUARES	41-42	
27	7/11/24	PYRAMID OF NUMBERS	43-44	

28	7/11/24	CHARACTER FREQUENCY IN A STRING	45	
29	7/11/24	ADD 'ING' AT THE END OF A STRING	46	
30	7/11/24	RETURN LENGTH OF LONGEST WORD	47-48	
31	14/11/24	PATTERN PRINTING	49-50	
32	14/11/24	FACTORS OF A NUMBER	51-52	
33	14/11/24	AREA OF SQUARE, RECTANGLE, TRIANGLE USING LAMBDA FUNCTION	53-54	
34	21/11/24	WORKING WITH GRAPHIC PACKAGES	55-58	
35	22/11/24	COMPARE AREA OF 2 RECTANGLE OBJECTS	59-60	
36	22/11/24	BANK ACCOUNT TRANSACTION USING CLASS	61-63	
37	29/11/24	OPERATOR OVERLOADING	64-65	
38	29/11/24	DISPLAY INFORMATION ABOUT A PYTHON BOOK	66-68	
39	6/12/24	READ FILE LINE BY LINE	69	
40	6/12/24	COPY ODD LINES OF FILE TO ANOTHER	70-71	

Experiment No:1

FUTURE LEAP YEARS FROM THE CURRENT YEAR TO FINAL YEAR

AIM

To write a Python program to display future leap years from the current year to final year.

ALGORITHM

1. Take current_year and final_year as input from the user
2. If final_year is less than or equal to current_year, display an error and stop.
3. Create an empty list to store leap years.
4. Loop through all years from current_year to final_year.
 - Add a year to the list if it is divisible by 4 and not 100, or divisible by 400.
5. Display the list of leap years.

PROGRAM

```
current_year = int(input("Enter the current year:"))
final_year = int(input("Enter the final year: "))
list2=[]
if final_year <= current_year:
    print("The final year must be greater than the current year.")
else:
    print("Leap Years from", current_year, "to", final_year, "are:")
    for year in range(current_year, final_year + 1):
        if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
            list2.append(year)
    print(list2)
```

OUTPUT

```
Enter the current year:2024
Enter the final year: 2051
Leap Years from 2024 to 2051 are:
[2024, 2028, 2032, 2036, 2040, 2044, 2048]
```

RESULT

The program to display future leap yeas from the current year to final year has executed and verified successful

Experiment No: 2

GENERATE POSITIVE LIST OF NUMBERS FROM A LIST

AIM

To write a Python program generate positive list of numbers from a given list of numbers.

ALGORITHM

1. Create two lists
2. Enter n integers into a list and print the list
3. If list element greater than 0, add to the second list
4. Print the second list

PROGRAM

```
list1 = []
list2 = []
n = int(input("Enter the limit: "))
print("Enter %d integers: " % n)
for i in range(0, n):
    a = int(input())
    list1.append(a)
print("List of integers: ", list1)
for i in range(0, n):
    if list1[i] > 0:
        b = list1[i]
        list2.append(b)
print("List of positive integers: ", list2)
```


OUTPUT

```
Enter the limit: 5
Enter 5 intergers:
1
-101
5
6
-78
List of integers: [1, -101, 5, 6, -78]
List of positive integers: [1, 5, 6]
```

RESULT

The program to generate positive list of numbers from a given list of numbers had executed and verified successfully

Experiment No: 3

SQUARE OF N NUMBERS

AIM

Write a python program to find the square of N numbers.

ALGORITHM

1. Create 2 lists
2. Enter n elements into a list and print
3. Calculate square of numbers in the list
4. Append the squares into second list
5. Print the list of squares

PROGRAM

```
list1 = []
list2 = []
n = int(input("Enter the limit: "))
print("Enter %d numbers:" % n)
for i in range(0, n):
    a = int(input())
    list1.append(a)
print("List of numbers: ", list1)
for i in range(0, n):
    b = list1[i] ** 2
    list2.append(b)
print("List of squares: ", list2)
```

OUTPUT

```
enter a range:5  
List of squares: [1, 4, 9, 16, 25]  
|
```

RESULT

The program to find the square of N numbers executed and verified successfully.

Experiment No: 4

FORM A LIST OF VOWELS OF A WORD

AIM

Write a python program to form a list of vowels selected from a given word.

ALGORITHM

1. Input a word
2. Compare each letter of the word with vowels
3. For all letters, if letter is a vowel
4. Add the letter to a list
5. Print list

PROGRAM

```
s=input("Enter a word:")
list2=[]
for i in s:
    if i=='a' or i=='e' or i=='i' or i=='o' or i=='u':
        list2.append(i)
print(list2)
```

OUTPUT

```
Enter a word:hello  
List of vowels: ['e', 'o']
```

RESULT

The program to form a list of vowels selected from a given word verified and executed successfully.

Experiment No: 5

LIST ORDINAL VALUE OF EACH ELEMENT OF A WORD

AIM

Write the python program to list ordinal value of each element of a word (HINT: use ord() to get ordinal values).

ALGORITHM

1. Enter a word
2. For each letter in the word, find the ordinal value
3. Add the ordinal values to a list
4. Print the list

PROGRAM

```
s=input("Enter a string:")
list2=[]
for i in s:
    t=ord(i)
    list2.append(t)
print(list2)
```

OUTPUT

```
Enter a string:python  
[112, 121, 116, 104, 111, 110]
```

RESULT

The program to list ordinal value of each element of a word verified and executed successfully.

Experiment No: 6

OCCURRENCE OF EACH WORD

AIM

Write the python program to count the occurrences of each word in a line of text.

ALGORITHM

1. Take input for a line of text and a word.
2. Split the text into a list of words.
3. Initialize a counter to zero.
4. Loop through the list of words.
5. If the word matches, increment the counter.
6. Print the count of the word.

PROGRAM

```
string=input("Enter the line of text:")
word=input("Enter the word:")
a=[]
a=string.split(" ")
count=0
for i in range(0,len(a)):
    if(word==a[i]):
        count=count+1
print("Count of the word is:")
print(count)
```


OUTPUT

```
Enter the line of text:python is simple to learn and python is dynamic programmi
ng
Enter the word:python
Count of the word is:
2
```

RESULT

The program to count the occurrences of each word in a line of text verified and executed successfully.

Experiment No: 7

LIST OF INTEGERS

AIM

Write the program to prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

ALGORITHM

1. Create a list
2. Input n integers into the list
3. If integer in list greater than 100, replace with 'over'
4. Print the updated list

PROGRAM

```
list1 = []
n = int(input("Enter the limit: "))
print("Enter the elements: ")
for i in range(0, n):
    a = int(input())
    list1.append(a)
print("List of integers: ", list1)
for i in range(0, n):
    if list1[i] > 100:
        list1[i] = "over"
print("Final list: ", list1)
```

OUTPUT

```
Enter the limit: 5
Enter the elements:
101
200
34
56
109
List of integers: [101, 200, 34, 56, 109]
Final list: ['over', 'over', 34, 56, 'over']
```

RESULT

The program for all values greater than 100, store 'over' instead by taking the values from the user verified and executed successfully.

Experiment No: 8

COUNT THE OCCURRENCES

AIM

Write the python program to store a list of first names. Count the occurrences of 'a' within the list.

ALGORITHM

1. Create a list
2. Enter a list of first names into the created list
3. For each names in the list, check the characters
4. If character in name is 'a', increment count
5. Print count as count of 'a'

PROGRAM

```
names = input("Enter first names separated by spaces: ").split()
a_count = 0
for name in names:
    a_count += name.lower().count('a')
print("Total occurrences of 'a':", a_count)
```

OUTPUT

```
Enter first names separated by spaces: Arya Anagha Riya Parvathy
Total occurrences of 'a': 8
```

RESULT

The program to store a list of first names, count the occurrences of 'a' within the list verified and executed successfully.

Experiment No- 9

LIST COMPARISON

AIM

Write the python program enter 2 lists of integers. Check

- (A) Whether lists are of same length
- (B) Whether list sums to same value
- (C) Whether any value occur in both

ALGORITHM

1. Create 2 lists
2. Enter integers into both lists
3. Compare the length of two lists
4. Compare the sum of elements of two lists
5. Display the common elements in both lists

PROGRAM

```
list1 = list(map(int, input("Enter the integers for the first list separated by spaces: ").split()))
list2 = list(map(int, input("Enter the integers for the second list separated by spaces: ").split()))
if len(list1) == len(list2):
    print("Both lists are of the same length.")
else:
    print("The lists are of different lengths.")
if sum(list1) == sum(list2):
    print("Both lists sum to the same value.")
else:
    print("The lists do not sum to the same value.")
common_values = set(list1) & set(list2)
if common_values:
    print("Common values found in both lists:", common_values)
else:
    print("No common values in both lists.")
```

OUTPUT

```
Python3: C:\Users\Aruna\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Aruna\AppData\Local\Programs\Python\Python39\pgm10.py
Enter the integers for the first list separated by spaces: 10 20 30 40 50
Enter the integers for the second list separated by spaces: 5 6 7 8 9 10
The lists are of different lengths.
The lists do not sum to the same value.
Common values found in both lists: {10}
```

RESULT

The program to check the lists are of same length, list sums to same value, any value occur in both lists are verified and executed successfully.

Experiment No: 10

OCCURRENCE OF FIRST CHARACTER REPLACED

AIM

Write the python program get a string from an input string where all occurrences of first character replaced with '\$', except first character.

ALGORITHM

1. Take input for a string.
2. Store the first character of the string.
3. Replace all occurrences of the first character with '\$'.
4. Concatenate the first character with the modified string.
5. Print the result.

PROGRAM

```
name=input("Enter a string:")  
s=name[0]  
name=name.replace(s,'$')  
print(s+name[1:])
```


OUTPUT

```
Enter a string: onion
oni$n
```

RESULT

The program get a string from an input string where all occurrences of first character replaced with '\$', except first character is verified and executed successfully.

Experiment No: 11

FIRST AND LAST CHARACTERS EXCHANGE

AIM

Write the python program create a string from given string where first characters exchanged.

ALGORITHM

1. Enter a word
2. Find the length
3. Print the last character first, then middle characters then the first character

PROGRAM

```
n=input("Enter a string:")  
s=n[-1]+n[1:-1]+n[0]  
print("The string:",s)
```

OUTPUT

```
Enter a string:Python  
The string: nythoP
```

RESULT

The program to create a string from given string where first characters exchanged verified and executed successfully.

Experiment No-12

AREA OF CIRCLE

AIM

Write python program accept radius from user and find the area of circle.

ALGORITHM

- 1.import math package
- 2.Enter the radius of circle
- 3.find the area of circle
- 4.Print area

PROGRAM

```
import math
r=int(input("Enter the radius of the circle:"))
area=math.pi*r*r
print("Area of the circle=",area)
```

OUTPUT

```
Enter the radius of the circle:2
Area of the circle= 12.566370614359172
```

RESULT

The program accept radius from user and find the area of circle verified and executed successfully.

Experiment No-13

BIGGEST OF THREE NUMBERS

AIM

Write a Python program to accept three numbers from user and find the biggest number.

ALGORITHM

1. Read three numbers from user.
2. Convert these values into float.
3. Store these values in three variables num1, num2, num3.
4. If num1 is greater than num2 and num3 then print num1 is largest.
5. If num2 is greater than num1 and num3 then print num2 is largest.
6. Else print num3 is largest.

PROGRAM

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))
if a>b and a>c:
    largest=a
elif b>a and b>c:
    largest=b
else:
    largest=c
print("Largest=",largest)
```

OUTPUT

```
Enter the first number:10  
Enter the second number:20  
Enter the third number:30  
Largest= 30
```

RESULT

Biggest of three numbers is executed and verified successfully.

Experiment No:14

TO PRINT FILENAME EXTENSION

AIM

To write a Python program to print the filename extension.

ALGORITHM

1. Enter a file name with extension
2. Split filename by '.' and add it to a list
3. Display the last element in the list

PROGRAM

```
a=input("Enter the file name:")  
s=a.split('.')  
print(s[-1])
```

OUTPUT

```
Enter the file name:number.py  
py
```

RESULT

The program to print the filename extension has executed and verified successfully.

Experiment No-15

DISPLAY FIRST AND LAST COLORS FROM LIST OF COLORS

AIM

Create a list of colors from comma-separated color names entered by user .Display first and last colors.

ALGORITHM

- 1.Enter a comma separated list of colors
- 2.Split colors and add to a list
- 3.Display the color at first index
- 4.Display the color at last index

PROGRAM

```
colors = input("Enter color names separated by commas: ").split(',')  
print("First color:", colors[0])  
print("Last color:", colors[-1])
```

OUTPUT

```
Enter color names separated by commas: blue,pink,green,yellow  
First color: blue  
Last color: yellow
```

RESULT

The program to display first and last colors from list of colors is executed and verified successfully.

Experiment No: 16

INTEGER OPERATIONS

AIM

Write a python program to accept an integer n and compute $n+nn+nnn$.

ALGORITHM

1. Accept an integer n from the user.
2. Calculate the square of n ($n \times n$).
3. Calculate the cube of n ($n \times n \times n$).
4. Add n, nn, and nnn to get the final result.
5. Display the result to the user.

PROGRAM

```
n=int(input("Enter the number:"))
nn=n*n
nnn=n*n*n
result=n+nn+nnn
print("The sum of n+nn+nnn=",result)
```

OUTPUT

```
Enter the number:5
The sum of n+nn+nnn= 155
```

RESULT

The program to perform integer operation has executed and verified successfully.

Experiment No:17

PRINT COLORS IN COLOR-LIST1 NOT CONTAINED IN COLOR-LIST2

AIM

Write a python program to print out all colors from color-list1 not contained in color-list2.

ALGORITHM

1. Create 2 list of colors color-list1 and color-list2
2. Print the 2 list of colors
3. Display color-list1 excluding colors in color-list2

PROGRAM

```
list_1 = set(["White", "Black", "Red"])
list_2 = set(["Red", "Green"])
print("Original set elements:")
print(list_1)
print(list_2)
print("\nDifference of list_1 and list_2:")
print(list_1.difference(list_2))
```

OUTPUT

```
Original set elements:
{'Red', 'Black', 'White'}
{'Red', 'Green'}

Difference of list_1 and list_2:
{'Black', 'White'}
```

RESULT

The program to print out all colors from color-list1 not contained in color-list2 has executed and verified successfully.

Experiment No-18

GENERATE SINGLE STRING FROM TWO STRINGS BY SWAPPING THE CHARACTER AT POSITION 1

AIM

Write a python program to create a single string separated with space from two strings by swapping the character at position 1.

ALGORITHM

1. Enter first string
2. Enter second string
3. Replace first of string1 with first character of string2
4. Replace first of string2 with first character of string1
5. Display the 2 updated strings separated by space

PROGRAM

```
str1 = input("Enter First String: ")
str2 = input("Enter Second String: ")
str3 = str1.replace(str1[0], str2[0], 1)
str4 = str2.replace(str2[0], str1[0], 1)
print(str3 + " " + str4)
```

OUTPUT

```
Enter First String: Learning
Enter Second String: python
pearning Lython
```

RESULT

The program to create a single string separated with space from two strings by swapping the character at position 1 has executed and verified successfully.

Experiment No-19

SORT DICTIONARY IN ASCENDING AND DESCENDING.

AIM

Write a python program to sort dictionary in ascending and descending order.

ALGORITHM

1. Create a dictionary
2. Sort and print dictionary in ascending order
3. Sort and print dictionary in descending order

PROGRAM

```
names = {4:'Anu' ,1:'Arya' ,6:'Dev' ,2:'Lena' ,3:'Ravi' ,6:'Debi' ,5:'Harry'}  
print("Ascending order: ",sorted(names.items()))  
print("Descending order: ",sorted(names.items(),reverse="TRUE"))
```

OUTPUT

```
Ascending order: [(1, 'Arya'), (2, 'Lena'), (3, 'Ravi'), (4, 'Anu'), (5, 'Harry'), (6, 'Debi')]  
Descending order: [(6, 'Debi'), (5, 'Harry'), (4, 'Anu'), (3, 'Ravi'), (2, 'Lena'), (1, 'Arya')]
```

RESULT

The program to Sort dictionary in ascending and descending order has executed and verified successfully.

Experiment No:20

MERGE TWO DICTIONARIES

AIM

Write a python program to merge two dictionaries.

ALGORITHM

1. Create dictionary1 with values
2. Create dictionary2 with values
3. Display two dictionaries
4. Merge and display 2 dictionaries

PROGRAM

```
dict1={"Apple":20,"Orange":30,"Grapes":40}  
dict2={"Blueberry":10,"Litchi":30}  
dict1.update(dict2)  
print("Merged dictionary:",dict1)
```

OUTPUT

```
Merged dictionary: {'Apple': 20, 'Orange': 30, 'Grapes': 40, 'Blueberry': 10, 'Litchi': 30}
```

RESULT

The program to merge two dictionaries has executed and verified successfully.

Experiment No-21

GCD OF TWO NUMBERS

AIM

Write a Python program to find gcd of 2 numbers.

ALGORITHM

1. Input two numbers a and b.
2. While $b \neq 0$, repeat steps 3 and 4.
3. Calculate the remainder of a divided by b.
4. Set $a=b$ and $b = \text{remainder}$.
5. Output a as the GCD.

PROGRAM

```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
while b != 0:
    temp = b
    b = a % b
    a = temp
print(f"The GCD of the numbers is {a}")
```

OUTPUT

```
Enter the first number: 1575
Enter the second number: 231
The GCD of the numbers is 21
```

RESULT

The program to calculate the gcd of 2 numbers has executed and verified successfully.

Experiment No-22

REMOVE EVEN NUMBERS FROM A LIST

AIM

From a list of integers, create a list removing even numbers.

ALGORITHM

1. Create a list
2. Enter n integers into the list
3. If (integer % 2) not equal to 0
4. Add integer to second list
5. Display the second list

PROGRAM

```
n=int(input("enter the number of elements in the list:"))
a=[]
b=[]
print("Enter a list of %d integers"%(n))
for i in range(n):
    element=int(input())
    a.append(element)
print("The list of integers:",a)
for i in a:
    if i%2!=0:
        b.append(i)
print("The list after removing even numbers:",b)
```

OUTPUT

```
enter the number of elements in the list:6
Enter a list of 6 integers
21
22
23
24
25
26
The list of integers: [21, 22, 23, 24, 25, 26]
The list after removing even numbers: [21, 23, 25]
```

RESULT

The program to create a list removing even numbers from a list of integers has executed and verified successful

Experiment No-23

FACTORIAL OF A NUMBER

AIM

To write a Python program to accept a number from user and find the factorial of the number

ALGORITHM

1. Read an input from user
2. Convert it into integer
3. Initialize a variable to one
4. If input is less than zero then print an error message
5. If input is equal to zero then display 1 as output
6. Else calculate factorial and print the result
7. Exit.

PROGRAM

```
num = int(input("Enter the number to find factorial: "))
factorial = 1
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1, num + 1):
        factorial = factorial * i
print("The factorial of the given number:", factorial)
```


OUTPUT

```
Enter the number to find factorial:5  
The factorial of the given number: 120
```

RESULT

The program to find the factorial of a number has executed and verified successfully.

Experiment No-24

FIBONACCI SERIES

AIM

To write a Python program to generate the Fibonacci series of n terms.

ALGORITHM

1. Input the 'n' value until which the Fibonacci series has to be generated
2. Initialize sum = 0, f1 = 0, f2 = 1 and count = 1
3. while (count <= n)
4. Print sum
5. Increment the count variable
6. Swap f1 and f2
7. while (count > n)
8. Else
9. Repeat from steps 4 to 7

PROGRAM

```
n = int (input ("Enter the number of n terms: "))
f1 = 0
f2 = 1
sum = 0
count = 1
print ("Fibonacci Series of n terms:",end = " ")
while (count <= n):
    print (sum, end = " ")
    count += 1
    f1 = f2
    f2 = sum
    sum = f1 + f2
```

OUTPUT

```
Enter the number of n terms: 10
Fibonacci Series of n terms: 0 1 1 2 3 5 8 13 21 34
|
```

RESULT

The program to generate Fibonacci series of n terms has executed and verified successfully.

Experiment No-25

SUM OF ALL ITEMS IN A LIST

AIM

Find the sum of all item in a list.

ALGORITHM

1. Create a list
2. Enter n numbers into the list
3. Calculate the sum of numbers in the list
4. Print sum

PROGRAM

```
n=int (input("Enter the number of elements in the list: "))
list1= []
print ("Enter %d elements: "%(n))
for i in range (0, n):
    a=int (input ())
    list1.append(a)
print("list is",list1)
print("The sum of elements in the list",sum(list1))
```

OUTPUT

```
Enter the number of elements in the list: 5
Enter 5 elements:
10
20
30
40
50
list is [10, 20, 30, 40, 50]
The sum of elements in the list 150
```

RESULT

The program to find the sum of all item in a list has executed and verified successfully.

Experiment No-26

PRINT THE PERFECT SQUARES

AIM

To write a Python program to print the perfect squares upto the limit accepted from user

ALGORITHM

1. Loop through numbers i from 1000 to 9999.
2. Check if i is a perfect square.
3. If i is a perfect square, extract its digits.
4. Check if all digits are even.
5. If all digits are even, print i.

PROGRAM

```
import math
for i in range(1000,10000):
    num=int(math.sqrt(i))
    if(num*num==i):
        n=i
        while n!=0:
            r=n%10
            n=n//10
            if r%2!=0:
                break
        else:
            print(i)
```

OUTPUT

```
4624  
6084  
6400  
8464
```

RESULT

The program to print prefect squares has executed and verified successfully.

Experiment No-27

PYRAMID OF NUMBERS

AIM

Write a python program to print a given pyramid of numbers.

ALGORITHM

1. Read the step size n from the user.
2. Start a loop from $i = 1$ to n (inclusive).
3. For each i , start an inner loop from $j = 1$ to i (inclusive).
4. In the inner loop, print k and increment it by i after each print.
5. After completing each iteration of the inner loop, print a new line.

PROGRAM

```
n=int(input("Enter the step size:"))
```

```
for i in range(1,n+1):
```

```
    k=i
```

```
    for j in range(1,i+1):
```

```
        print(k,end=' ')
```

```
        k+=i
```

```
    print()
```


OUTPUT

```
Enter the step size:5
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

RESULT

The program to print the pyramid given size has executed and verified successfully.

Experiment No-28

CHARACTER FREQUENCY IN A STRING

AIM

Count the number of characters in a string.

ALGORITHM

1. Enter a string
2. Calculate frequency of each character
3. Print each character's frequency

PROGRAM

```
str1 = input("Enter a string: ")
for i in str1:
    frequency = str1.count(i)
    print(str(i) + ":" + str(frequency), end=",")
```

OUTPUT

```
Enter a string: programming
p:1,r:2,o:1,g:2,r:2,a:1,m:2,m:2,i:1,n:1,g:2,
```

RESULT

The program to count the number of characters in a string has executed and verified successfully.

Experiment No-29

ADD “ING”AT THE END OF THE STRING.

AIM

Add “ing” at the end of the given string.If it already ends with “ing”,then add “ly”.

ALGORITHM

- 1.Enter a string
- 2.if string end with 'ing', add 'ly'
- 3.Else, add 'ing'

PROGRAM

```
string = input ("Enter a string: ")  
  
if string [-3:] == 'ing':  
    print (string + 'ly')  
  
else:  
    print (string + 'ing')
```

OUTPUT

```
Enter a string: walk  
walking  
|
```

RESULT

The program to add ‘ing’ at the end of a string has executed and verified successfully.

Experiment No-30

RETURN LENGTH OF LONGEST WORD.

AIM

Accept a list of word and return length of longest word.

ALGORITHM

1. Create a list
2. Enter n elements into the list
3. Set length of first list element as max
4. Compare max with the list elements and find max size
5. Print the max

PROGRAM

```
list1 = []
n = int(input("Enter the number of elements in the list: "))
print("Enter %d elements" % (n))
for i in range(0, n):
    a = input()
    list1.append(a)
max1 = len(list1[0])
temp = list1[0]
for i in list1:
    if len(i) > max1:
        max1 = len(i)
        temp = i
print("Length of longest word:", len(temp))
```

OUTPUT

```
Enter the number of elements in list:5
Enter 5 elements
HAI
HELLO
HOW
WELCOME
BYE
Length of longest word: 7
```

RESULT

The program to accept a list of word and return length of longest word has executed and verified successfully.

Experiment No-31

PATTERN PRINTING

AIM

To write a Python program to construct the following pattern using nested for loop

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

ALGORITHM

1. Take the number n (step size) from the user.
2. Print increasing “*” starting from 1 to n.
3. Print decreasing “*” starting from n-1 down to 1.
4. The program completes after printing the full pattern.

PROGRAM

```
n=int(input("enter the step size:"))
for i in range(1,n+1):
    print('* '*i)
for i in range(n-1,0,-1):
    print('* '*i)
```

OUTPUT

```
enter the step size:5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

RESULT

The program to print the given pattern has executed and verified successfully.

Experiment No-32

FACTORS OF A NUMBER

AIM

To write a Python program to accept a number from user and generate all factors of that number.

ALGORITHM

1. Read an input from user and pass the value to the function
2. Use a for loop to iterate i from 1 to given number
3. check if $\text{number \% } i == 0$ if true print the i else increment the loop value by 1 until i.
4. Exit

PROGRAM

```
num = int(input("Enter a number: "))  
  
print(f"Factors of {num} are:")  
  
for i in range(1, num + 1):  
    if num % i == 0:  
        print(i)
```


OUTPUT

```
Enter a number: 68
Factors of 68 are:
1
2
4
17
34
68
```

RESULT

The program to find factorial of a given number has executed and verified successfully.

Experiment No-33

AREA OF SQUARE, RECTANGLE, AND TRIANGLE USING LAMBDA FUNCTION

AIM

Write lambda function to find area of square, rectangle, and triangle.

ALGORITHM

1. Define lambda function for finding area of square
2. Define lambda function for finding area of rectangle
3. Define lambda function for finding area of triangle
4. Calculate and display area of square
5. Calculate and display area of rectangle
6. Calculate and display area of triangle

PROGRAM

```
s=lambda a:a*a
r=lambda x,y:x*y
t=lambda b,h:(b*h)/2

a=int(input("enter the side of a square:"))
print("the area of the square is:",s(a))

x=int(input("enter the length of the rectangle :"))
y=int(input("enter the breadth of the rectangle:"))
print("the area of the rectangle is",r(x,y))

b=int(input("enter the base of the triangle :"))
h=int(input("enter the height of the triangle :"))
print("the area of the triangle is",t(b,h))
```

OUTPUT

```
enter the side of a square:4
the area of the square is: 16
enter the length of the rectangle :4
enter the breadth of the rectangle:2
the area of the rectangle is 8
enter the base of the triangle :10
enter the height of the triangle :5
the area of the triangle is 25.0
|
```

RESULT

The program to find area of square,rectangle,and triangle using Lambda function has executed and verified successfully

Experiment No: 34

WORKING WITH GRAPHIC PACKAGES

AIM

Write a python program to find area and perimeter of figures by different importing statements.

ALGORITHM

1. Create a package called graphics.
2. Include modules rectangle and circle
3. Include methods to find area and perimeter of rectangle and circle.
4. Create a sub package called three graphics.
5. Include modules cuboid and sphere.
6. Include methods to find area and perimeter of rectangle and circle.
7. Import statements to print the area and perimeter of figures.
8. Exit.

PROGRAM

circle.py

```
print("*****CIRCLE**") import math
def circle_perimeter(radius):
    return math.pi * radius * 2
def circle_area(radius):
    return math.pi * radius * radius
def main():
    radius = float(input("Enter value for radius of circle: "))
    area = circle_area(radius)
    perimeter = circle_perimeter(radius)
    area = round(area, 4)
    perimeter = round(perimeter, 4)
    print("The Area of the circle is:", area)
    print("The perimeter of the circle is:", perimeter)
main()
```

rectanle.py

```
print("*****RECTANGLE**")
def rect_perimeter(l, w):
    return 2 * (l + w)
def rect_area(l, w):
    return l * w
def main():
    l = int(input("Enter the length of rectangle: "))
    w = int(input("Enter the width of rectangle: "))
    area = rect_area(l, w)
    perimeter = rect_perimeter(l, w)
    area = round(area, 4)
    perimeter = round(perimeter, 4)
    print("Area of rectangle:", area)
    print("Perimeter of rectangle:", perimeter)
main()
```

cuboid.py

```
print("*****CUBOID*****")
def cuboid_SA(length, width, height):
    return 2 * (length * width + height * width + length * height)
def cuboid_volume(length, width, height):
    return length * width * height
def main():
    length = float(input("Please enter the length of the cuboid: "))
    width = float(input("Please enter the width of the cuboid: "))
    height = float(input("Please enter the height of the cuboid: "))
    area = cuboid_SA(length, width, height)
    volume = cuboid_volume(length, width, height)
    area = round(area, 4)
    volume = round(volume, 4)
    print("The Surface Area of the cuboid is:", area)
    print("The Volume of the cuboid is:", volume)
main()
```

sphere.py

```
print("****SPHERE****")
import math
def sphere_area(radius):
    return 4 * math.pi * radius ** 2
def sphere_vol(radius):
    return (4 / 3) * math.pi * radius ** 3
def main():
    radius = float(input("Enter the radius of the sphere: "))
    area = sphere_area(radius)
    volume = sphere_vol(radius)
    area = round(area, 4)
    volume = round(volume, 4)
    print("The area of the sphere is:", area)
    print("The perimeter of the sphere is:", volume)
main()
```

main.py

```
import graphics.circle
import graphics.rectangle
from graphics.threedgra.cuboid import*
from graphics.threedgra.sphere import*
```

OUTPUT



```
Python 3.12.1
File Edit Shell Debug Options Window Help
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\DEVI\py_learn\CO3\main.py
****CIRCLE****
Enter value for radius of circle :3
The Area of circle is : 28.2743
The perimeter of circle is : 18.8496
****RECTANGLE****
Enter the length of rectangle:4
Enter the width of rectangle:2
Area of rectangle : 8
perimeter of rectangle : 12
****CUBOID****
please enter the length of a cuboid:6
please enter the width of a cuboid:2
please enter the height of a cuboid:3
The Area of cuboid is : 42.0
The perimeter of cuboid is : 36.0
****SPHERE****
enter radius of sphere:2
The area of sphere is: 50.2655
The perimeter of sphere is: 33.5103
>>>
```

RESULT

The python program to find area and perimeter of figures by different importing statements.

Experiment No-35

COMPARE AREA OF TWO RECTANGLE OBJECTS

AIM

Create Rectangle class with attributes length and breadth and methods to find area and perimeter.
Compare two rectangle objects by their area.

ALGORITHM

1. Create a class rectangle with attributes length and breadth to find area and perimeter of a rectangle.
2. Create three function inside class Rectangle.
 - i. def int () for inputing length & breadth.
 - ii. def area() to find area of the rectangle.
 - iii. def perimeter() to find perimeter of a rectangle
3. Display the result of area of both rectangles and find it whether it is matching or not.
4. Display the result.

PROGRAM

```
class rect:
    def __init__(self,l,b):
        self.length=l
        self.width=b
    def findArea(self):
        b=(self.length * self.width)
        return b
    def findPerimeter(self):
        c=2*(self.length+self.width)
        return c
a1=rect(10,3)
a2=rect(1,3)
print("Area of rectangle1: ",a1.findArea())
print("Perimeter of rectangle1: ",a1.findPerimeter())
print("Area of rectangle2: ",a2.findArea())
```



```
print("Perimeter of rectangle2: ",a2.findPerimeter())  
if a1.findArea()>a2.findArea():  
    print("\nRectangle1 is greater")  
else:  
    print("\nRectangle2 is greater")
```

OUTPUT

```
Area of rectangle1: 30  
Perimeter of rectangle1: 26  
Area of rectangle2: 3  
Perimeter of rectangle2: 8  
  
Rectangle1 is greater  
|
```

RESULT

The program to find area of rectangle and perimeter of executed and verified successfully.

Experiment No-36

BANK ACCOUNT TRANSACTION USING CLASS

AIM

To create a bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

ALGORITHM

1. Create a class Bank_Account
2. Define a function `__init__()`
3. Define a function `deposit()` for deposit functions
4. Define a function `withdraw()` for withdrawing money
5. Define a function `display()` to display the balance amount after transaction

PROGRAM

```
class bank_account:
    def getData (self,name,accno,acctype,balance):
        self.name=name
        self.accno=accno
        self.acctype=acctype
        self.balance=balance
    def displayCustomer (self):
        print("customer Name:", self.name)
        print("Account Number:",self.accno)
        print("Account Type:", self.acctype)
        print("Balance amount :", self.balance)
    def deposit(self,amount):
        self.balance=self.balance+amount
        print("Account balance:", self.balance)
    def withdrawal(self,amount):
        if self.balance-amount<0:
            print("insufficient funds")
        else:
            self.balance=self.balance-amount
```

```

        print("Amount withdrawn")
        print("Balance amount:",self.balance)
print("Hello Welcome to BANKING SYSTEM")
ch=0
obj=bank_account()
while ch!=5:
    print("Select your option")
    print("1.New customer")
    print("2.Deposit")
    print("3.Withdrawal")
    print("4.Display")
    print("5.Exit")
    ch=int(input("Enter your choice:"))
    if ch==1:
        obj=bank_account()
        n=input("Enter name:")
        no=int(input("Enter account number:"))
        t=input("Enter Account Type(SB/C):")
        b=int(input("Enter the Amount:"))
        obj.getData(n,no,t,b)
    if ch==2:
        b=int(input("Enter the amount to be deposited :"))
        obj.deposit(b)
    if ch==3:
        b=int(input("Enter the amount to be withdrawns:"))
        obj.withdrawal(b)
    if ch==4:
        obj.displayCustomer()
else:
    print("program terminated !!!")

```

OUTPUT

```
Hello Welcome to BANKING SYSTEM
Select your option
1.New customer
2.Deposit
3.Withdrawal
4.Display
5.Exit
Enter your choice:1
Enter name:RICHU MARIYAM ALEX
Enter account number:123405
Enter Account Type(SB/C):C
Enter the Amount:50000
program terminated !!!
Select your option
1.New customer
2.Deposit
3.Withdrawal
4.Display
5.Exit
Enter your choice:4
customer Name: RICHU MARIYAM ALEX
Account Number: 123405
Account Type: C
Balance amount : 50000
Select your option
1.New customer
2.Deposit
3.Withdrawal
4.Display
5.Exit
Enter your choice:2
Enter the amount to be deposited :1000
Account balance: 51000
```

RESULT

The program to create bank account and display the transaction has executed and verified successfully.

Experiment No-37

OPERATOR OVERLOADING

AIM

Create a class rectangle with private attributes length and breadth. Overload '<' operator to compare the area of two rectangles.

ALGORITHM

1. Create a class rectangle with attributes length and breath
2. Create two functions inside the class
 - a) define __int__ for inputing length and breadth
 - b) define __lt__ which overload the '<' operator to compare and display area of 2 rectangle objects
3. Enter length and breadth of two rectangle and find the area
4. Compare the area and display which is smallest

PROGRAM

```
class rectangle:
    def __init__(self):
        self.length=int(input("Enter length: "))
        self.breadth=int(input("Enter breadth: "))
    def __lt__(self,ob2):
        area1=self.length * self.breadth
        area2=ob2.length *ob2.breadth
        print("the area1:",area1)
        print("the area2:",area2)
        return (area1<area2)

print("Rectangle-1")
og1=rectangle()
print("Rectangle-2")
og2=rectangle()
if og1<og2:
```

```
        print("rectangle 1 is smaller")
    else:
        print("rectangle 2 is smaller")
```

OUTPUT

```
Rectangle-1
Enter length: 6
Enter breadth: 2
Rectangle-2
Enter length: 7
Enter breadth: 3
the area1: 12
the area2: 21
rectangle 1 is smaller
```

RESULT

The program to overload '<' operator has executed and verified successfully.

Experiment No: 38

DISPLAY INFORMATION ABOUT A PYTHON BOOK

AIM

To create a class publisher (name).derive class book from publisher with attribute title and author. Derive class python from book with attribute price and no: of pages. Display information about a python book .

ALGORITHM

1. Create a Publisher class with a display() method.
2. Create a Book class inheriting from Publisher and add title and author attributes.
3. Create a Python class inheriting from Book and add price and pages attributes.
4. Take input for publisher, title, author, price, and pages.
5. Create an object of the Python class.
6. Call the display() method to print all details.

PROGRAM

```
class Publisher:
    def __init__(self, name):
        self.name = name
    def display(self):
        print("Publisher name:", self.name)
class Book(Publisher):
    def __init__(self, name, title, author):
        super().__init__(name)
        self.title = title
        self.author = author
    def display(self):
        super().display()
        print("Book title:", self.title)
        print("Author name:", self.author)
class Python(Book):
    def __init__(self, name, title, author, price, pages):
```

```
        super().__init__(name, title, author)
        self.price = price
        self.pages = pages
    def display(self):
        super().display()
        print("Price:", self.price)
        print("Number of pages:", self.pages)
publisher_name = input("Enter the publisher name: ")
book_title = input("Enter the book title: ")
author_name = input("Enter the author's name: ")
book_price = float(input("Enter the book price: "))
number_of_pages = int(input("Enter the number of pages: "))
print()
book = Python(publisher_name, book_title, author_name, book_price, number_of_pages)
book.display()
```


OUTPUT

```
Enter the publisher name: python
Enter the book title: introduction to python
Enter the author's name: jeeva jose
Enter the book price: 800
Enter the number of pages: 500

Publisher name: python
Book title: introduction to python
Author name: jeeva jose
Price: 800.0
Number of pages: 500
```

RESULT

The program to class publisher and display information about a python book has executed and verified successfully.

Experiment No: 39

READ FILE LINE BY LINE

AIM

To write a Python program to read a file line by line and store it in a list.

ALGORITHM

1. Create a function named “read_file(file_name)”
2. Create a text file named file1.txt for writing the content
3. Read the file file1.txt and store it in the list and display the output.

PROGRAM

```
def read_file(file_name):  
    with open(file_name) as f:  
        content_list=f.readlines()  
    print(content_list)  
read_file('file1.txt')
```

INPUT TEXT FILE: file1.txt

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

OUTPUT

```
["Python is an interpreted, object-oriented, high-level programming language  
with dynamic semantics. Python's simple, easy to learn syntax emphasizes read  
ability and therefore reduces the cost of program maintenance.\n"]
```

RESULT

The program to print the filename extension has executed and verified successfully.

Experiment No: 40

COPY ODD LINES OF ONE FILE TO ANOTHER

AIM

To write a python program to copy odd lines of one text file to another text file

ALGORITHM

1. Create a file named file1.txt and file2.txt
2. Write some data line by line in file1.txt file
3. file2.txt file is used to store odd lines from file1.txt file
4. After copying the odd lines to file2.txt file display the content of file2.txt file

PROGRAM

```
file1=open('file1.txt','r')
file2=open('file2.txt','w')
lines=file1.readlines()
type(lines)
for i in range(0, len(lines)):
    if(i % 2 == 0):
        file2.write(lines[i])
file1 = open('file1.txt', 'r')
file2 = open('file2.txt', 'r')
str1 = file1.read()
str2 = file2.read()
print("file1 content...")
print(str1)
print()
print("file2 content...")
print(str2)
file1.close()
file2.close()
```

OUTPUT

```
file1 content...  
aa  
bb  
cc  
dd  
  
file2 content...  
aa  
cc
```

RESULT

The python program to copy odd lines of one text file to another text file has executed and verified successfully.

