

Experiment - 1

Aim:

Write a program to perform string manipulation operations using set of pre-defined functions such as :

1. `find()`

2. `upper()`

3. `len()`

4. `max()` and `min()`

5. Fetching a specific content from the string

10

Software Used: Python IDLE

Theory:

15 A string is a sequence of characters and can contain letters, numbers, symbols or spaces in quotation mark. String manipulation - It is a process of manipulating and analyzing strings.

(1) `find()` - searches the string for a specified value and returns the position of where it was found.

(2) `upper()` - converts the string into upper case

(3) `len()` - tells the length of string

(4) `max()` - returns the max alphabetical character from string.

25 `min()` - returns the min alphabetical character from string

(5) `split()` - splits the string at the specified separator and return a list

Teacher's Signature:

Output:

Enter a string: "HELLOWORLD" ✓
Enter the element to find: "W" ✓
The input element is at index = 5 ✓
String in uppercase = HELLOWORLD ✓
Length of string = 10 ✓
max alphabet = W ✓
min alphabet = D ✓
['HELLO WORLD'] ✓

code:

```
string = str(input("Enter a string:"))
element = input("Enter the element to find:")
find = string.find(element)
print("The input element is at index =", find)
print("String in upper case =", string.upper())
print("Length of string =", len(string))
print("Max alphabet =", max(string))
print("Min alphabet =", min(string))
sp = string.split() # Fetching a specific content by converting it in
print(sp)
```

Explanation:

In the above code we have used the following pre-defined functions in python i.e. find(), upper(), len(), max(), min() and split() whose ~~work~~ and hence manipulate the strings.

Sushant
02/05/23

20

25

Output:

Enter a number: 18.45

number rounded off up to nearest integer of = 13

Enter a number to find its square root: 25

Square root of 25 = 5.0

Enter a number: 2

Enter another number: 3

2 raise to the power 3 = 8.0

Enter a number to print its factorial: 5

factorial of 5 = 120

Experiment - 2

AIM:

Write a program to perform a test and check mathematical functions such as

1. ceil()

2. sqrt()

3. pow()

4. factorial()

10. Software used: Python IDLE

Theory:

1. math.ceil - It rounds a number up to the nearest integer.
2. math.sqrt - Returns the square root of a number.
3. math.pow(x, y) - Returns the value of x to power y
4. math.factorial() - Returns the factorial of Number.

Code:

```

20 import math
num = int(input('Enter a number:'))
print('Number rounded off to the nearest integer of', num, '=', math.
      math.ceil(num))
num2 = int(input('Enter a number to find its square root'))
print('Square root of', num2, '=', math.sqrt(num2))
x = int(input('Enter a number'))
y = int(input('Enter another number'))

```

```
print('raise to the power:', y, '=', math.pow(x, y))
```

```
num3 = int(input('Enter a number to print its factorial'))
```

```
print('Factorial of ', num3, '=', math.factorial(num3))
```

Explanation:

mathematical

In the above code, we have used predefined functions under the ~~math~~ library. That allows us to perform mathematical tasks on numbers. ~~Please Note~~

Output:

Enter a number: 12

The entered number is even

Experiment - 3

Aim

5 write a program that receive a number as input from user and return if it odd or even number.

Software used

Jupyter Notebook

Theory

The if else statement is used to execute a code block among two alternatives.

If the condition is true, the block of code under the if statement will be executed

15 If the condition is false, the block of code under the else statement will be executed

Syntax

~~if (condition):~~

statement to be executed if condition is TRUE

20 ELSE:

Statement to be executed otherwise

Code

num = int(input('Enter a Number'))

25 if (num % 2 == 0):

print('The entered Number is even')

ELSE:

print('The entered Number is odd')

Explanation

In the above code, we have used if - else statement. We have to check whether the entered number is even or odd so the condition is if the number is divisible by 2 it will be even and not divisible by 2 it will be odd. So we have given the condition under if statement that whether the entered number is divisible by 2 and if the number is divisible by 2 statement under if ie 'The entered number is even' gets printed.

Experiment - 4

Aim

5 write a program that receives an input from the user to calculate the area of triangle

Software used: Jupyter Notebook

Theory:

10 operators are used to perform operations on variables and values

Arithmetic operators

Arithmetic operators are used with numeric values to perform common mathematical operations

15 + Addition $x + y$

- Subtraction $x - y$

* Multiplication $x * y$

/ Division x / y

% Modulus $x \% y$

** Exponential $x^{**} y$

// Floor division $x // y$

The area of triangle can be calculated with the given sides with the help of Heron's formula given by $\sqrt{s(s-a)(s-b)(s-c)}$ where $s = a+b+c/2$ and $a, b,$ and c are sides.

Output

Enter first side: 3

Enter second side: 4

Enter third side: 5

The area of triangle $^{\circ}$ is 6.

code

```
a = int(input('Enter first side:'))  
b = int(input('Enter second side:'))  
c = int(input('Enter third side:'))  
s = (a+b+c)/2  
A = (s*(s-a)*(s-b)*(s-c))**0.5  
print('The area of triangle is', A)
```

Explanation:

After taking input from user 3, 4 and 5 the computer calculates the value of s as $\frac{3+4+5}{2} = 6$.
and area comes as 6.

Output

Enter the side of square: 12

The area of square is 144.0 sq unit

Experiment - 5

Aim

write a program to receive input from the user to calculate the area of square.

5

software used: ~~Jupyter notebook~~

Theory :

Arithmetic operations are used ~~for~~ with numeric values

to perform basic mathematical function.

python math module is an important feature designed to deal with mathematical operations

math.pow(x, y) → returns the value of x to power y

Code

import math.

side = float(input('Enter the side of square:'))

area = math.pow(side, 2)

print(f'the area of square is, {area}, "sq unit")

20

Explanation

as the area of square is $side^2$ we have entered the side as 12 unit ~~for~~, our program has shown

output as 144 sq unit as ~~we~~ has used predefined math function (math.pow()) under math module. which has returned the value as side raise to power 2.

Output

Enter length of rectangle: 11

Enter Breadth of rectangle: 12

The area of rectangle is 132.0 sq unit

Experiment - 6

Aim

write a program to receive input from user
to calculate area of rectangle.

5

software used Jupyter Notebook

Theory

Arithmetic operations are used with numeric values
to perform basic mathematical operation
for multiplying two numbers, a and b we write $a * b$
code

```
l = float(input('Enter length of rectangle:'))  
b = float(input('Enter breadth of rectangle:'))  
area = l * b  
print(f'Area of rectangle = {area} sq unit')
```

Explanation

In the above code, we have entered length as 12
unit and breadth as 11 unit and the output
is shown as 132 sq unit ie 11×12 . The computer has
performed multiplication (11 and 12) and shown the
output ie area of rectangle.

25

Enter a string: Nitin

Yes, the entered string is palindrome.

Experiment - 7

Aim

Write a program that takes input from user and check the given string is palindrome or not.

Software used - Jupyter Notebook

Theory

A palindrome means a word, or sentence that reads the same backward or forward.

Code

```
str = input('Enter a string')
```

```
If str[::] == str[::-1]:
```

~~print ('Yes, the entered string is palindrome')~~

else:

~~print ('No, the entered string is not palindrome')~~

Explanation

We have used string slicing here. It means obtaining a sub-string from the given string by slicing it.

~~str[::]~~ means to display the first to last element

~~and str[::-1]~~ means to display the first to last elements in reverse order. If both are equal the string is

said to be palindrome.

Enter marks of student : 98

B

Experiment 8

Aim

write a program that receives marks of student
as input and assign grades A | B | C | D | E | F

5

Software used

Jupyter Notebook

Theory

- If statement is used to run a block of code only when a certain condition under if is true.
If the given condition is false, the statement under else is executed.
Elif statement allows to check for the multiple condition

Code

```
marks = float(input('Enter marks of student'))
```

```
if marks > 90:
```

```
    print('A')
```

```
elif marks <= 90 or marks > 80:
```

```
    print('B')
```

```
elif marks <= 80 or marks > 70:
```

```
    print('C')
```

```
elif marks <= 70 or marks > 60:
```

```
    print('D')
```

```
elif marks <= 60 or marks > 50:
```

Teacher's Signature:

Experiment Name / No.: _____

Camlin / Page No.

Date / /

```
print('E')
```

```
Else :
```

```
print ('F')
```

Explanation

If - Else - Else statement is used here, If the marks of student is above 90, the grade ~~sho~~ will be shown as A, If marks is between 90 and 80 grade will be B, and If else statements are used here based on the conditions.

Very good

Sribhav
30/05/23

Output:

Enter a number: 34

Enter second number: 12

2

Experiment - 9

Aim

Write a program to compute GCD of two numbers.

Software used

Jupyter notebook

Theory

The greatest common divisor (GCD) of two or more numbers is the greatest common factor number that divides them exactly.

Code:

```

x = int(input('Enter a number'))
y = int(input('Enter second number'))
if x < y:
    num = x
else:
    num = y
for i in range(1, num + 1):
    if x % i == 0 and y % i == 0:
        hcf = i
print(hcf)

```

Explanation

here, we have input two numbers x and y , and check for the smaller one. If x is smaller then, num will be equal to x .

Experiment Name / No.: _____

Camlin / Page No.

Date / /

and if y smaller, num will be equal to y. Then there is
a for loop to check for the factors of x and y if there
is a common factor ie a number divisible by both
x and y and that is highest, this will be considered and
printed as highest common factor

5

10

Experiment - 10

Aim

Write a program to check if the given number is Armstrong
or not

Theory

An Armstrong number is a number in which the sum of digits raised to power number of digits is equal to the original digit.

for example

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

code

15 import math

num = int(input('Enter a number'))

num1 = num

num2 = num

count = 0

20 while(num != 0):

 item = num % 10

 quo = num // 10

 num = quo

 count = count + 1

25 print(count)

sum = 0

while (num1 != 0):

Output

Enter a number 370

The entered number is Armstrong number

rem1 = num1 % 10

quot1 = num1 // 10

num1 = quot1

5 p = math.pow(rem1, count)

Sum = sum + p

print(sum)

if num2 == sum:

print('The entered number is a Armstrong number')

else:

10 print('The entered number is not a Armstrong number')

Explanation

Two while loops are used here, first one to count the number of digits and second one to sum up the digits

15 raised to power (no of digit). We have entered a number say 370, then it will pass through while loop 1. This loop will be executed 3 times as long as $370 \neq 0$. hence the value of count will be 3, which is equal to no of digits in 370. In second loop the individual digit of 20 370 will be raised to power 3 and added, $(3)^3 + (7)^3 + (0)^3 = 27 + 343 + 0 = 370$, hence it is an Armstrong number.

*Syham
13/06/23*

Enter a year to check whether it's leap or not: 2016

The entered year is a leap year.

Experiment - 11

Aim

Write a program to check if the input year is leap year or not.

Software used : Jupyter Notebook

Theory

The if statement checks the conditions provided, it displays the output based on a given condition. else statement prints the output of the given condition is false.

Code :

```
x = int(input('Enter a year to check whether its leap or not'))  
if ((x % 4 == 0) and (x % 100 == 0) or (x % 100 != 0) and (x % 4 == 0)):  
    print ('The entered year is leap year')  
else:
```

```
    print ('The entered year is not leap year')
```

Explanation

The above block of code check whether the entered year is leap or not. If the year entered is divisible by both 4 and 100 or it is divisible by 4 only then it will be considered as leap year.

Output

Enter a number to print factorial: 5

120 is the factorial of number

Experiment - 18

Aim:

write a program to compute factorial of a number

5

Software used: Jupyter Notebook

Theory

while loop:

10 while loop statement in python repeatedly executes a
the statement as long as the given condition is true

Code:

15 $n = \text{int}(\text{input}(\text{Enter a number to print its factorial}))$. $p = 1$ $i = 1$ while $i \leq n$: $p = p * i$ $i = i + 1$ 20 $\text{print}(p, ' is the factorial of number')$ Explanation: we have taken a number as input here
and assigned (p) and (i) the value & we used while loop
as long as ~~the value of i becomes greater than or equal to n~~ value of i equal to n .
The statement will execute as long as i is smaller than
or equal to n , and compute $(p) \times (i)$ and hence will print factorial

Teacher's Signature

Yashika
13/06/23

Experiment Name / No.: _____

Output:

how many numbers? 10

0

1

1

2

3

5

8

13

21

34

55

o
Aim

o
Write a program

o Software used

Theory

Fibonacci series

following sequence

in which a

numbers

python will print

a given condition

while condition

statement

Experiment-13

Aim:

Write a program to generate fibonacci series till 100

Software used : Jupyter notebook

Theory

Fibonacci series numbers are the numbers in the following sequence - 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

in which a number is the sum of two previous numbers

python while loop is used to run a block of code until a given condition is true.

while (condition):

 Statement to be executed

Code:

a = 0

b = 1

while n = int(input('How many numbers?')):

 i = 1

 while i < n:

 c = a + b

 print(c)

 a = b

 b = c

Explanation

In the above code initially the value of $a = 0$ and $b = 1$ by using while loop we can print further series. Initially the value of i is 1 and loop will print value until the value become equal to input value (n). then c is the sum of a and b and value of a will be equal to b and b will be equal to c then further value of c will change accordingly and hence fibonacci series will be printed till ~~previous~~ value of ' n '.

Output

enter a number 10

$$10 \times 1 = 10$$

$$10 \times 2 = 20$$

$$10 \times 3 = 30$$

$$10 \times 4 = 40$$

$$10 \times 5 = 50$$

$$10 \times 6 = 60$$

$$10 \times 7 = 70$$

$$10 \times 8 = 80$$

$$10 \times 9 = 90$$

$$10 \times 10 = 100$$

Experiment - 14

Ques.

Write a program to print a multiplication table of a given number.

5

Software used Jupyter Notebook

Theory

for loop in python is used to iterate over sequence such as list, tuple & strings

Syntax:

for [iterate in sequence]:
 statement

15 Code

```
n = int(input('Enter a number'))  
for i in range(1,11):  
    print(n, 'x', i, '=', n*i)
```

20 Explanation

In the above code, ~~all~~ 'n' is number of entered while loop is ~~repeated~~ run 10 times in range (1,11) and print output which is ~~n*i~~. The print statement is executed 10 times.

25

Experiment - 15

Aim

Write a program to create two list and perform.

- 1 Add elements of two list
- 2 compare the contents of list
- 3 to find number of elements in list
- 4 sort elements of list
- 5 reverse the contents of list

Software used Jupyter Notebook

Theory

list are used to store multiple items in a single variable.
list items are indexed. one first item has index[0] and second has [1]. list are mutable i.e we can change values inside of list.

Code:

```
def addlist(list1, list2):  
    result = []  
    for i in range(len(list1)):  
        result.append(list1[i] + list2[i])  
    return result  
  
def compare(list1, list2):  
    if len(list1) == len(list2):  
        print("Lengths are equal")  
    else:  
        print("Lengths are not equal")  
  
def count(list1):  
    print("Length of list is", len(list1))
```

[7, 9, 11, 13, 15]

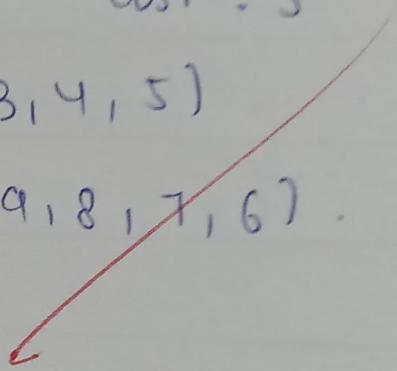
Different

not equal

No. of elements in list : 5

Sorted list = [1, 2, 3, 4, 5]

Reversed list = [10, 9, 8, 7, 6]



```
def sort_list():
    return sorted(list)

def reverse_list():
    return list[:: -1]

list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]

sum_list = add_list(list1, list2)

if compare(list1, list2):
    print("Equal")
else:
    print("not Equal")

count = count(list2)
print("no of elements in list:", count)

sorted_list = sort_list(list2)
print("sorted list:", sorted_list)

reverse_list = reverse(list2)
print("reversed list:", reverse_list)
```

Explanation:

This program defines functions for each operation and uses them to perform the required tasks. It creates two lists 'list1' and 'list2', and then calls the function to add elements, compare lists, find no of elements, sort the list and reverse are printed according to the function.

Output

city Name

New York

London

Paris

TOKYO

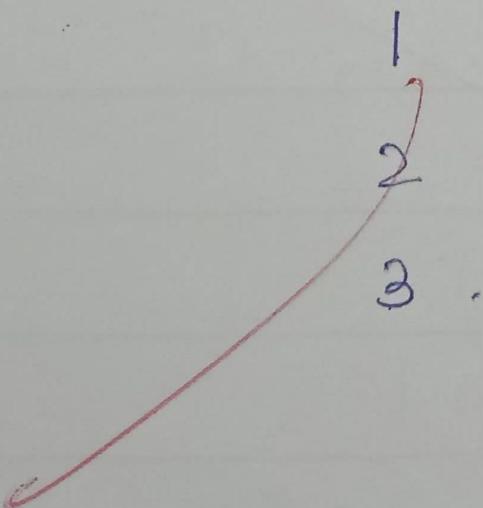
Index position

0

1

2

3



Experiment-16

Aim

Write a program to create and display the content of tuple initializing the tuple with the name of the cities. Display content of tuple along with name / index position of cities.

Software used : Jupyter Notebook.

10

Theory

Tuples are used to store multiple items in a single variable. It is basically a collection of data separated by comma and enclosed within brackets. Python tuple is immutable.

15

Code :

```
def display_cities(cities):
    print('city name at index position')
    print('1/n')
    for index, city in enumerate(cities):
        print(f'{index} {city}')
cities = ('New York', 'London', 'Paris', 'Tokyo')
display_cities(cities)
```

Explanation

In the main part of program a tuple 'cities' is created with a few city names. The display_cities function is

Teacher's Signature:

Experiment Name / No.: _____

Camlin / Page No.

Date / /

called with this tuple as argument, which prints the
content of tuple along with name | index.

5

~~Surbhi~~
~~27/06/23~~

INDEX

Sr. No.	Experiment	Page	Date of Experiment	Date of Submission	Remarks
1	WAP to perform string manipulation		18 April 23	8 May 23	?
2	WAP to perform mathematical func.		25 April 23	8 May 23	AT
3	WAP to check whether a number is even or odd		2 May 2023	17 May	
4	WAP to calculate area of triangle		2 May 2023	17 May	
5	WAP to calculate area of square		14 May 2023	30 May	
6	WAP to calculate area of Rectangle		17 May 2023	30 May	
7	WAP to check whether input string is palindrome		14 May 2023	30 May	
8	WAP to input marks and assign grades		14 May 2023	30 May	?
9	WAP to compute GCD of two no		30 May 2023	13 June	AT
10	WAP to check if given no is Armstrong or not.		30 May 2023	13 June	AT
11	WAP to check input year is leap or not		13 June	13 June	
12	WAP to compute factorial of number		13 June	13 June	?
13	WAP to generate Fibonacci series		20 June	27 June	AT
14	WAP to generate multiplication table		20 June	27 June	AT
15	WAP to create two list and perform different functions.		27 June	27 June	AT
16	WAP to create & display content of tuple.		27 June	27 June	