

PRACTICAL I

DATE: 08/01/24

Basic programs in python

AIM: To learn and execute Basic programs in python.

INTRODUCTION:

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions: Python 2 and Python 3. Both are quite different.

LANGUAGE FEATURES

Interpreted:

- There are no separate compilation and execution steps like C and C++.
- Directly run the program from the source code.
- Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.
- No need to worry about linking and loading with libraries, etc.

Platform Independent:

- Python programs can be developed and executed on multiple operating system platforms.
- Python can be used on Linux, Windows, Macintosh, Solaris and many more.
- Free and Open Source; Redistributable

High-level Language:

- In Python, no need to take care about low-level details such as managing the memory used by the program.

Simple:

- Closer to English language; Easy to Learn
- More emphasis on the solution to the problem rather than the syntax

Embeddable:

- Python can be used within C/C++ program to give scripting capabilities for the program's users.

Robust:

- Exceptional handling features
- Memory management techniques in built

Rich Library Support:

- The Python Standard Library is very vast.
- Known as the “batteries included” philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.
- Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

Pros:

- Ease of use
- Multi-paradigm Approach

Cons:

- Slow speed of execution compared to C,C++
- Absence from mobile computing and browsers
- For the C,C++ programmers switching to python can be irritating as the language requires proper indentation of code. Certain variable names commonly used like sum are functions in python. So C, C++ programmers have to look out for these.

Advantages:

- Presence of third-party modules
- Extensive support libraries(NumPy for numerical calculations, Pandas for data analytics etc)
- Open source and community development
- Versatile, Easy to read, learn and write
- User-friendly data structures
- High-level language
- Dynamically typed language(No need to mention data type based on the value assigned, it takes data type)
- Object-oriented language
- Portable and Interactive
- Ideal for prototypes – provide more functionality with less coding
- Highly Efficient(Python’s clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient.)
- (IoT)Internet of Things Opportunities

- Interpreted Language
- Portable across Operating systems

Applications:

- GUI based desktop applications
- Graphic design, image processing applications, Games, and Scientific/ computational Applications
- Web frameworks and applications
- Enterprise and Business applications
- Operating Systems
- Education
- Database Access
- Language Development
- Prototyping
- Software Development

Variables and Data Structures

In other programming languages like C, C++, and Java, you will need to declare the type of variables but in Python you don't need to do that. Just type in the variable and when values will be given to it, then it will automatically know whether the value given would be an int, float, or char or even a String.

Python program to declare variables

```
myNumber = 3
```

```
print(myNumber)
```

```
myNumber2 = 4.5
```

```
print(myNumber2)
```

```
myNumber = "helloworld"
```

```
print(myNumber)
```

Python have 4 types of built in Data Structures namely List, Dictionary, Tuple and Set.

List is the most basic Data Structure in python. List is a mutable data structure i.e items can be added to list later after the list creation. It's like you are going to shop at the local market and made a list of some items and later on you can add more and more items to the list.

append() function is used to add data to the list.

Input and Output

input() function is used to take input from the user.

```
# Python program to illustrate
# getting input from user
name = input("Enter your name: ")
# user entered the name 'harssh'
print("hello", name)
```

Selection

Selection in Python is made using the two keywords 'if' and 'elif' and else (elseif)

```
# Python program to illustrate
# selection statement
num1 = 34
if(num1>12):
    print("Num1 is good")
elif(num1>35):
    print("Num2 is not gooooo....")
else:
    print("Num2 is great")
```

Functions

You can think of functions like a bunch of code that is intended to do a particular task in the whole Python script. Python used the keyword 'def' to define a function.

Syntax:

```
def function-name(arguments):
def hello():
    print("hello")
    print("hello again")
# calling function
```

```
hello()
```

Iteration (Looping)

As the name suggests it calls repeating things again and again. We will use the most popular 'for' loop here.

```
# Python program to illustrate  
# a simple for loop  
for step in range(5):  
    print(step)
```

Modules

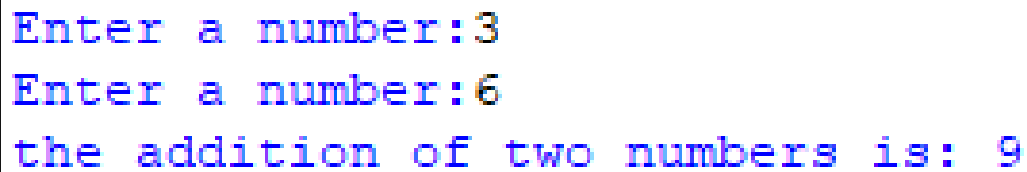
Python has a very rich module library that has several functions to do many tasks. You can read more about Python's standard library by [Clicking here](#)

'import' keyword is used to import a particular module into your python code. For instance consider the following program.

```
# Python program to illustrate  
# math module  
import math  
def Main():  
    num = -85  
    # fabs is used to get the absolute  
    # value of a decimal  
    num = math.fabs(num)  
    print(num)  
if __name__=="__main__":  
    Main()
```

Q1. Python program to add two numbers

```
a=int (input("Enter a number:"))  
b=int(input("Enter a number:"))  
print("the addition of two numbers is:",a+b)
```



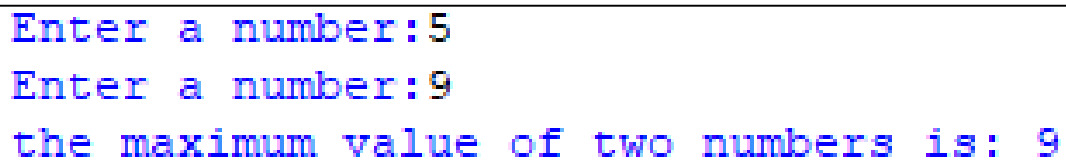
A screenshot of a terminal window showing the execution of a Python program. The text is displayed in a blue monospaced font. It shows two prompts 'Enter a number:' followed by the inputs '3' and '6', and finally the output 'the addition of two numbers is: 9'.

```
Enter a number:3  
Enter a number:6  
the addition of two numbers is: 9
```

Figure 1: Output of Q1

Q2. Python program to display Maximum of two numbers in Python

```
a=int (input("Enter a number:"))  
b=int(input("Enter a number:"))  
print("the maximum value of two numbers is:",max(a,b))
```



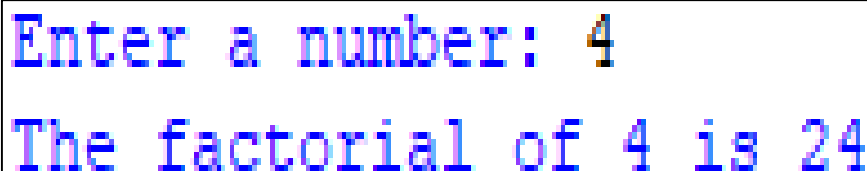
A screenshot of a terminal window showing the execution of a Python program. The text is displayed in a blue monospaced font. It shows two prompts 'Enter a number:' followed by the inputs '5' and '9', and finally the output 'the maximum value of two numbers is: 9'.

```
Enter a number:5  
Enter a number:9  
the maximum value of two numbers is: 9
```

Figure 2: Output of Q2

Q3. Python Program for factorial of a number

```
num = int(input("Enter a number: "))  
factorial = 1  
if num < 0:  
    print(" 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",num,"is",factorial)
```



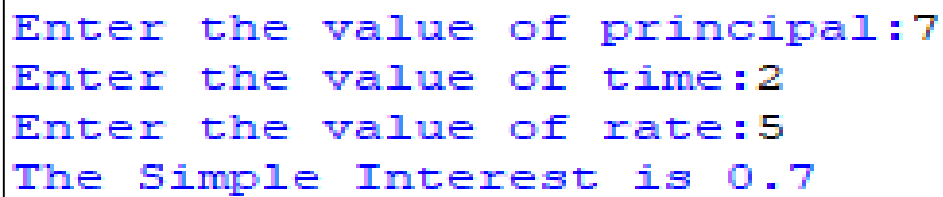
A screenshot of a terminal window showing the execution of a Python program. The text is displayed in a blue monospaced font. It shows a prompt 'Enter a number:' followed by the input '4', and finally the output 'The factorial of 4 is 24'.

```
Enter a number: 4  
The factorial of 4 is 24
```

Figure 3: Output of Q3

Q4. Python Program for simple interest

```
def simple_interest(p,t,r):  
  
    SI = (p * t * r)/100  
  
    print('The Simple Interest is', SI)  
    return SI;  
  
a=int(input ("Enter the value of principal:"))  
b=int(input ("Enter the value of time:"))  
c=int(input ("Enter the value of rate:"))  
simple_interest(a,b,c)
```

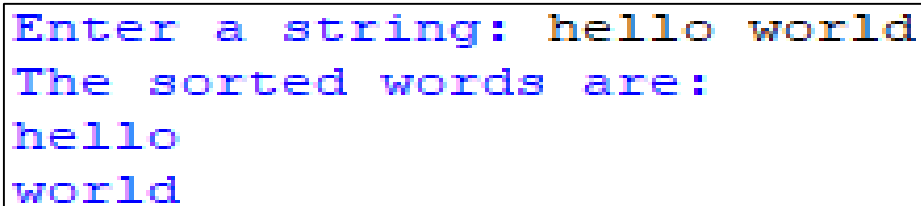


```
Enter the value of principal:7  
Enter the value of time:2  
Enter the value of rate:5  
The Simple Interest is 0.7
```

Figure 4: Output of Q4

Q5. Python Program to Sort Words in Alphabetic Order.

```
my_str = input("Enter a string: ")  
  
words = [word.lower()for word in my_str.split()]  
  
words.sort()  
print("The sorted words are:")  
for word in words:  
    print(word)
```



```
Enter a string: hello world  
The sorted words are:  
hello  
world
```

Figure 5: Output of Q5

Q6. Python Program to check Armstrong Number

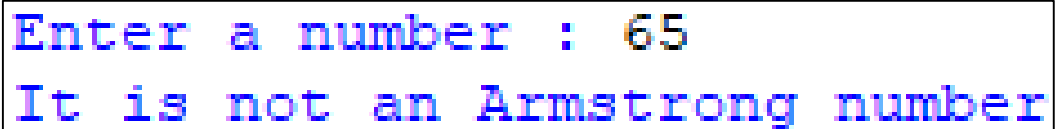
```
num = int(input("Enter a number : "))  
sum = 0  
temp = num  
  
while temp > 0:  
    digit = temp % 10
```

```

sum += digit * digit * digit
temp = temp//10

if sum==num:
    print("It is an Armstrong number")
else:
    print("It is not an Armstrong number")

```



Enter a number : 65
It is not an Armstrong number

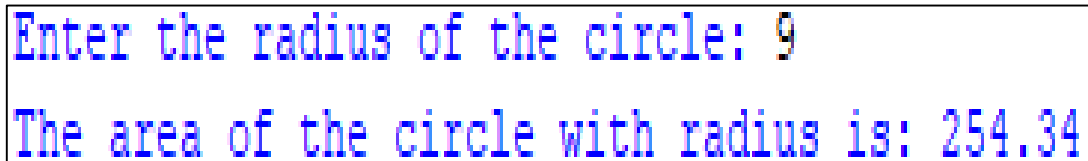
Figure 6: Output of Q6

Q7. Python Program for Program to find area of a circle

```

r = float(input("Enter the radius of the circle: "))
area = 3.14*r*r
print("The area of the circle with radius is:", area)

```



Enter the radius of the circle: 9
The area of the circle with radius is: 254.34

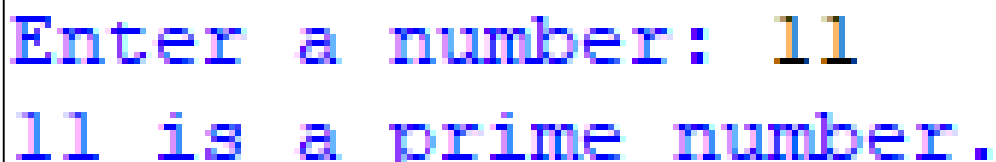
Figure 7: Output of Q7

Q8. Python program to check whether a number is Prime or not

```

num = int(input("Enter a number: "))
if num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            print(num, "is not a prime number.")
            break
    else:
        print(num, "is a prime number.")

```



Enter a number: 11
11 is a prime number.

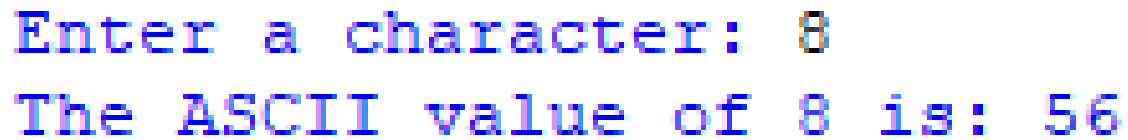
Figure 8: Output of Q8

Q9. Program to print ASCII Value of a character

```
character = input("Enter a character: ")

if len(character) == 1:
    ascii_value = ord(character)

    print("The ASCII value of " + character + " is: " + str(ascii_value))
else:
    print("Please enter a single character.")
```



```
Enter a character: 8
The ASCII value of 8 is: 56
```

Figure 9: Output of Q9

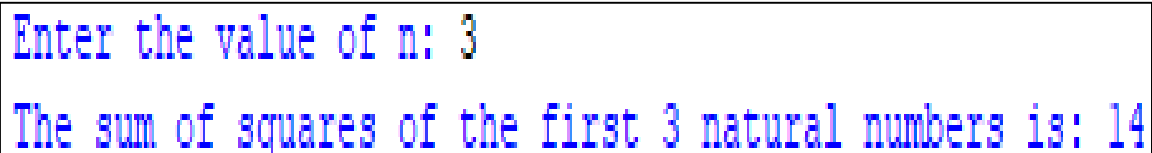
Q10. Python Program for Sum of squares of first n natural numbers

```
def sum_of_squares(n):

    sum_squares = 0
    for i in range(1, n + 1):
        sum_squares += i ** 2
    return sum_squares

n = int(input("Enter the value of n: "))

if n <= 0:
    print("Please enter a positive integer.")
else:
    answer = sum_of_squares(n)
    print("The sum of squares of the first", n, "natural numbers is:", answer)
```



```
Enter the value of n: 3
The sum of squares of the first 3 natural numbers is: 14
```

Figure 10: Output of Q10

Q11. Python Program for get a sequence from user and check whether it RNA or not

```
def is_RNA(sequence):
    for nucleotide in sequence:
        if nucleotide not in 'ACGU':
            return False
    return True
```

```
user_sequence = input("Enter the sequence to check if it's RNA: ")
```

```
if is_RNA(user_sequence.upper()):
```

```
    print("The sequence is RNA.")
```

```
else:
```

```
    print("The sequence is not RNA.")
```

```
Enter the sequence to check if it's RNA: aaauuagagc
The sequence is RNA.
```

Figure 11: Output of Q11