**First Internal Assessment Test:**

**Subject code and Name:GE8151, Problem solving and Python Programming**

**Date:13.10.2018 Marks:50**

**Time:10.00-11.30pm Duration:1.30hrs**

**PART-A:**

**Answer any five of the questions:**

1. **What is list? Difference between list and tuple**.

Ans. Lists are a container data type that acts as a dynamic array. That is to say, a list is a sequence that can be indexed into and that can grow and shrink. Lists are mutable, but tuples are *not.*

1. **What is dictionary with example?**

Ans. A dictionary is an unordered collection of key value pairs. A dictionary has a length, specifically the number of keyvalue pairs. A dictionary provides fast look up by key. The keys must be immutable object types.

vegetables = {

'Eggplant': 'Purple',

'Tomato': 'Red',

'Parsley': 'Green',

'Lemon': 'Yellow',

'Pepper': 'Green',}

1. **What are all the features of python programming language?**

Ans. ● Built in high level data types: strings, lists, dictionaries, etc.

● The usual control structures:

if, ifelse, ifelifelse, while,

plus a powerful collection iterator (for).

● Multiple levels of organizational structure: functions, classes, modules, and

packages. These assist in organizing code. An excellent and large example is the

Python standard library.

1. **What are the two basic modes of python?**

* Script mode
* Interactive mode

1. **Write a pseudo code for addition of two numbers.**

Ans. READ a ,b ,c

SET a=10,b=5

CALCULATE c=b + a

1. **What is recursive function?**

Ans. A recursive function is a function that calls itself.

A recursive function must have a limiting condition, or else it will loop endlessly.

Each recursive call consumes space on the function call stack. Therefore, the number of

recursions must have some reasonable upper bound.

**PART-B:**

**Answer any four questions:**

* 1. **Explain in detail about algorithm**.

Ans

|  |
| --- |
| *An****algorithm****is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time* [***[Levitin](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml" \l "Levitin)***, p. 3]. |

If the problem is that of computations, being "unambiguous" usually means that a human of average intelligence must be able (if only in principle) to follow the instructions with pencil and paper. In theory, a discerning robot must be able to perform the job as well. And this inescapably links the idea of algorithm to programming. Indeed, algorithms are commonly expressed either in a programming language like [***Pascal***](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml#Pascal) or [***C++***](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml#CPP), or a ***[pseudocode](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml" \l "Levitin)***. Some go an extra mile by inventing a custom programming language, like AL by [***Maurer and Ralston***](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml#DAM) and the notorious assembly for the hypothetical computer MIX by [***R. Knuth***](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml#Knuth). This is all for the purpose of making the Description of the algorithms as "unambiguous" as possible.

However, quite often, the concept of *algorithm* is thought to be distinct from that of *program*. The distinction is similar to that made in mathematics between the Platonic notion of *number* and a *magnitude* describing the size of a naturally available set (as in "three cows," say.)

Often, too, a program, unlike an algorithm, is not required to terminate in a finite time, and, indeed, the *Halting Problem*, that is the problem of existence of an algorithm that could predict whether a given program will terminate (in finite time, of course) for a given input, is a cornerstone of the *Computability Theory*.

In my view, the finiteness requirement is a red herring. Consider an example of typing out consecutive prime numbers. For any integer, there is an unambiguous (and finite) way to determine whether it is prime or not. We'll do this an integer at a time staring with, say, 2. We may set up a terminating condition (say, check the first million of integers or print out the first million of primes) or not. Compare

* 1. **Write an algorithm for sum of n numbers**.

Ans. 1.Read the value of n.  
2. i = 1 , SUM = 0  
3. if ( i > n ) go to 7  
4. S = S + i  
5. i = i + 1  
6. go to 3  
7. Display the value of S  
8. Stop

* 1. **Write notes on flowchart with its components**.

Ans. Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find less-obvious features within the process, like flaws and [bottlenecks](https://en.wikipedia.org/wiki/Bottleneck_(production)). There are different types of flowcharts: each type has its own set of boxes and notations. The two most common types of boxes in a flowchart are:

* a processing step, usually called *activity*, and denoted as a rectangular box.
* a decision, usually denoted as a diamond.

A flowchart is described as "cross-functional" when the chart is divided into different vertical or horizontal parts, to describe the control of different organizational units. A symbol appearing in a particular part is within the control of that organizational unit. A cross-functional flowchart allows the author to correctly locate the responsibility for performing an action or making a decision, and to show the responsibility of each organizational unit for different parts of a single process.

Flowcharts depict certain aspects of processes and are usually complemented by other types of diagram. For instance, [Kaoru Ishikawa](https://en.wikipedia.org/wiki/Kaoru_Ishikawa) defined the flowchart as one of the seven basic tools of quality control, next to the [histogram](https://en.wikipedia.org/wiki/Histogram), [Pareto chart](https://en.wikipedia.org/wiki/Pareto_chart), [check sheet](https://en.wikipedia.org/wiki/Check_sheet), [control chart](https://en.wikipedia.org/wiki/Control_chart), [cause-and-effect diagram](https://en.wikipedia.org/wiki/Ishikawa_diagram), and the [scatter diagram](https://en.wikipedia.org/wiki/Scatter_diagram). Similarly, in [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language), a standard concept-modeling notation used in software development, the [activity diagram](https://en.wikipedia.org/wiki/Activity_diagram), which is a type of flowchart, is just one of many different diagram types.

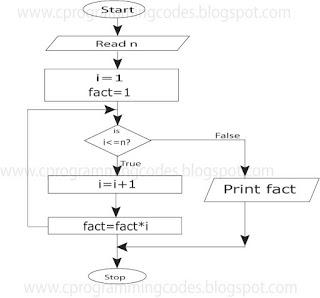
[Nassi-Shneiderman diagrams](https://en.wikipedia.org/wiki/Nassi-Shneiderman_diagram) and [Drakon-charts](https://en.wikipedia.org/wiki/DRAKON) are an alternative notation for process flow.

Common alternative names include: flow chart, process flowchart, functional flowchart, process map, process chart, functional process chart, business process model, process model, process [flow diagram](https://en.wikipedia.org/wiki/Flow_diagram), [work flow](https://en.wikipedia.org/wiki/Workflow) diagram, business flow diagram. The terms "flowchart" and "flow chart" are used interchangeably.

|  |  |  |
| --- | --- | --- |
| **ANSI/ISO Shape** | **Name** | **Description** |
| Flowchart Line.svg | Flowline (Arrowhead)[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Shows the process's order of operation. A line coming from one symbol and pointing at another.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) Arrowheads are added if the flow is not the standard top-to-bottom, left-to right.[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |
| Flowchart Terminal.svg | Terminal[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Indicates the beginning and ending of a program or sub-process. Represented as a [stadium](https://en.wikipedia.org/wiki/Stadium_(geometry)),[[14]](https://en.wikipedia.org/wiki/Flowchart" \l "cite_note-ShellyVermaat2011-14) oval or rounded (fillet) rectangle. They usually contain the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product". |
| Flowchart Process.svg | Process[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Represents a set of operations that changes value, form, or location of data. Represented as a [rectangle](https://en.wikipedia.org/wiki/Rectangle).[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |
| Flowchart Decision.svg | Decision[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Shows a conditional operation that determines which one of the two paths the program will take.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) The operation is commonly a yes/no question or true/false test. Represented as a diamond ([rhombus](https://en.wikipedia.org/wiki/Rhombus)).[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |

|  |  |  |
| --- | --- | --- |
| Flowchart IO.svg | Input/Output[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Indicates the process of inputting and outputting data,[[15]](https://en.wikipedia.org/wiki/Flowchart" \l "cite_note-Myler1998-15) as in entering data or displaying results. Represented as a [parallelogram](https://en.wikipedia.org/wiki/Parallelogram).[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) |
| Flowchart Annotation.svg | Annotation[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14)(Comment)[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) | Indicating additional information about a step the program. Represented as an open rectangle with a dashed or solid line connecting it to the corresponding symbol in the flowchart.[[15]](https://en.wikipedia.org/wiki/Flowchart#cite_note-Myler1998-15) |
| Flowchart Predefined Process.svg | Predefined Process[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Shows named process which is defined elsewhere. Represented as a rectangle with double-struck vertical edges.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) |
| Flowchart Connector.svg | On-page Connector[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14) | Pairs of labeled connectors replace long or confusing lines on a flowchart page. Represented by a small circle with a letter inside.[[14]](https://en.wikipedia.org/wiki/Flowchart#cite_note-ShellyVermaat2011-14)[[18]](https://en.wikipedia.org/wiki/Flowchart#cite_note-RFF-18) |

8.2 **Draw a flowchart for factorial of n numbers**.



9.1. **What is function? Write a python program for calculator with its basic operation using function.**

Ans. ● It groups a block of code together so that we can call it by name.

● It enables us to pass values into the the function when we call it.

● It can returns a value (even if None).

● When a function is called, it has its own namespace. Variables in the function are

local to the function (and disappear when the function exits). A function is defined with the def: statement

.

9.2 # Python program for simple calculator

  # Function to add two numbers

def add(num1, num2):

    return num1 + num2

  # Function to subtract two numbers

def subtract(num1, num2):

    return num1 - num2

  # Function to multiply two numbers

def multiply(num1, num2):

    return num1 \* num2

  # Function to divide two numbers

def divide(num1, num2):

    return num1 / num2

  print("Please select operation -\n" \

        "1. Add\n" \

        "2. Subtract\n" \

        "3. Multiply\n" \

        "4. Divide\n")

  # Take input from the user

select = input("Select operations form 1, 2, 3, 4 :")

  number\_1 = int(input("Enter first number: "))

number\_2 = int(input("Enter second number: "))

  if select == '1':

    print(number\_1, "+", number\_2, "=",

                    add(number\_1, number\_2))

  elif select == '2':

    print(number\_1, "-", number\_2, "=",

                    subtract(number\_1, number\_2))

  elif select == '3':

    print(number\_1, "\*", number\_2, "=",

                    multiply(number\_1, number\_2))

  elif select == '4':

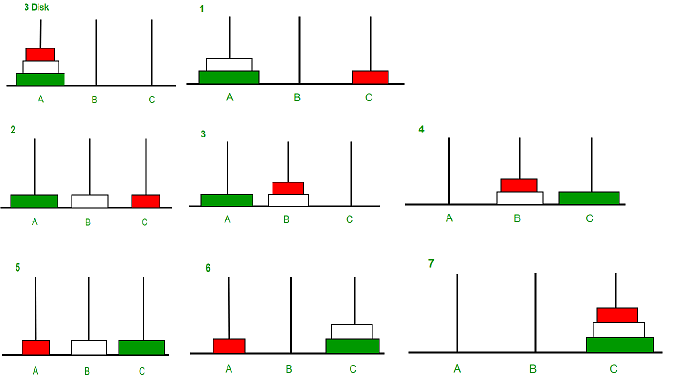
    print(number\_1, "/", number\_2, "=",

                    divide(number\_1, number\_2))

10.1. **What is tower of Hanoi? Solve tower of Hanoi problem with 3 number of disc**.

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:  
1) Only one disk can be moved at a time.  
2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.  
3) No disk may be placed on top of a smaller disk.

faq.disk3 faq.disk3 faq.disk3

faq.disk3

11.1. **Write a python program for Fibonacci series**.

Ans. # Program to display the Fibonacci sequence up to n-th term where n is provided by the user

# change this value for a different result

nterms = 10

# uncomment to take input from the user

#nterms = int(input("How many terms? "))

# first two terms

n1 = 0

n2 = 1

count = 0

# check if the number of terms is valid

if nterms <= 0:

print("Please enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence upto",nterms,":")

print(n1)

else:

print("Fibonacci sequence upto",nterms,":")

while count < nterms:

print(n1,end=' , ')

nth = n1 + n2

# update values

n1 = n2

n2 = nth

#count += 1

11.2. **Write a python program for factorial of given number.**

Ans. # Python program to find the factorial of a number provided by the user.

# change the value for a different result

num = 7

# uncomment to take input from the user

#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero

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",num,"is",factorial)