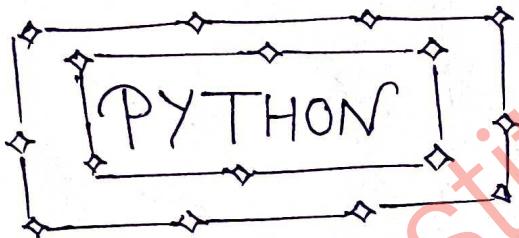


Python programming language

IIIrd Paper



By Rajesh Sir

Python Document PDF file
(Python programming language)

Python programming language

Chapter - 01, 02

* Introduction to Programming *

* Algorithm ! → एक Algorithm एक कार्य की पूरा करने या किसी समस्या की हल करने के लिये अच्छी तरह से परिचालित निर्देशों का एक क्रम है। Simple शब्दों में हम कह सकते हैं कि किसी Problem को Step-by-Step Solve करना। Algorithm कहलाता है।

Algorithm कोई भी अच्छी तरह से परिचालित Computational प्रक्रिया है, जो Input के रूप में कुछ Value या Value के set को लेती है, और output के रूप में कुछ Value या Value के set का उत्पादन करती है।

Example! → दो संख्याओं को जोड़ना।

Step 1! → BEGIN

Step 2! → Read a, b

Step 3! → c = a + b

Step 4! → Write c

Step 5! → END

* विशेषताएँ → इसकी निम्न विशेषताएँ हैं:

① Algorithm किसी भी Programming Language पर बिन्दी नहीं करती। इसलिये इस Programming Language से बदल प्रोग्राम होती है।

② यह सामान्य English Language है। जिसे Program के उद्देश्य को सही ढंग से के लिये Steps में लिखा गया है। जिसे अनासानी से समझा जा सकता है।

Python Document PDF file

(Python programming language)

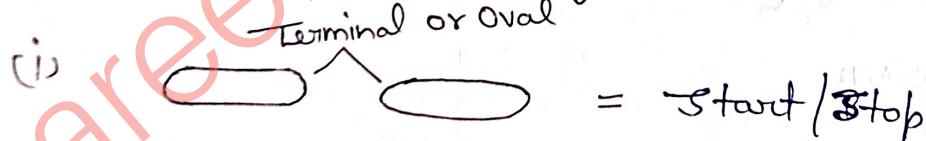
Python programming language

उ) यह और कोई Programming code नहीं है जो Program की किसी भी उन्नीशक read तथा write के लिये जटिलता मेंदा करता है।

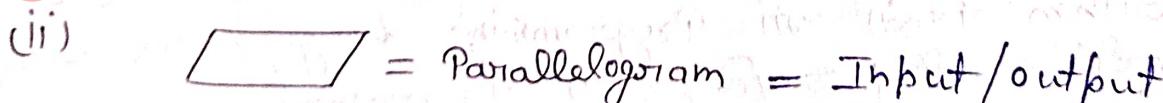
*. Algorithm के लाभ → इसके निम्नलिखित लाभ होते हैं :

- ① यह Programming Language से स्वतंत्र है।
- ② इसे समझना आसान होता है। क्योंकि यह किसी भी समस्या की step-by-step तरफ करती है।
- ③ इसे Debug करना आसान है। क्योंकि प्रत्येक step की अनुपना Logical sequence मिलता है।
- ④ महले एक Algorithm की develop करना आसान है। और फिर इसे एक flowchart में, फिर इसे एक Computer Program में परिवर्तित करते हैं।

*. Flowcharts : → Flowchart किसी Program की design करने का एक Graphical Representation होता है। इसकी स्थायता से हम किसी Program को solve करने का diagram तैयार करते हैं। इसमें विभिन्न प्रकार के symbols का use किया जाता है।



इसका use start तथा stop के लिये करते हैं।



(iii)  = Rectangle = Processing
Program में जो भी Numeric operation होता है उसे इसी box में लगा जाता है।

(iv)  = diamond/ decision box = Conditional box
 $(a > 10)$

Python programming language

(v)  = Flowchart = यह flowchart में जाकूती है। जिसका use program के Control flow को indicate करने के लिया किया जाता है।

*. Benefit of Flowchart →

- ① Flowchart की स्थायता से Problem का अच्छी तरह से विशलेषण किया जा सकता है। इसलिये यह सागत और समय की बचती कम होती है।
- ② Program Flowchart एक अच्छी Program documentation के रूप में काम करता है। जिसकी उन्नत विधि उद्देश्य के लिये होती है। जिससे वीजे आणि कुशल बनती है।
- ③ Flowchart System Analysis और Program development ~~one phase~~ के द्वारा एक Guide के रूप में काम करता है।
- ④ Flowchart debugging process में help करता है।
- ⑤ Flowchart की स्थायता से operating Program maintenance आसान होता है।

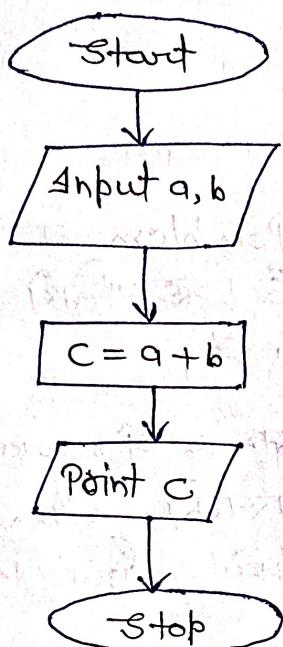
*. Limitation of Flowchart to use !

- ① कमी - कमी Program Logic काफी जटिल हो जाता है। इस Case में flowchart जटिलता अद्दा हो जाता है।
- ② यदि परिवर्तन आवश्यक है, तो flowchart की पूरी तरह से Redrawing की आवश्यकता होती है।
- ③ Flowchart के Symbols की Type नहीं किया जा सकता। इसलिये flowchart का Reproduction एक समस्या बन जाता है।

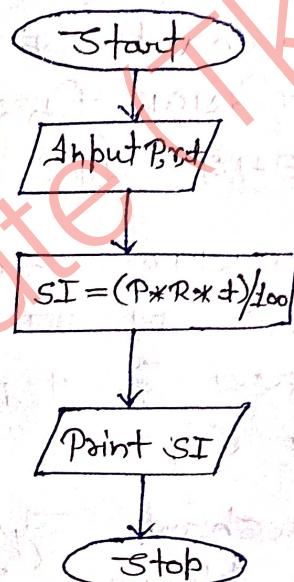
Python programming language

* Examples of Flowchart →

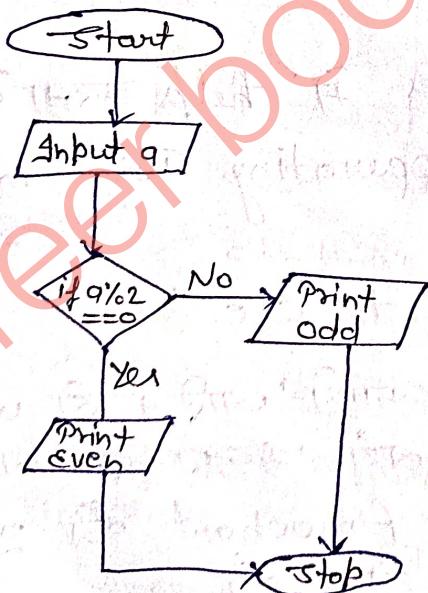
① दो Numbers को जोड़ने का Program



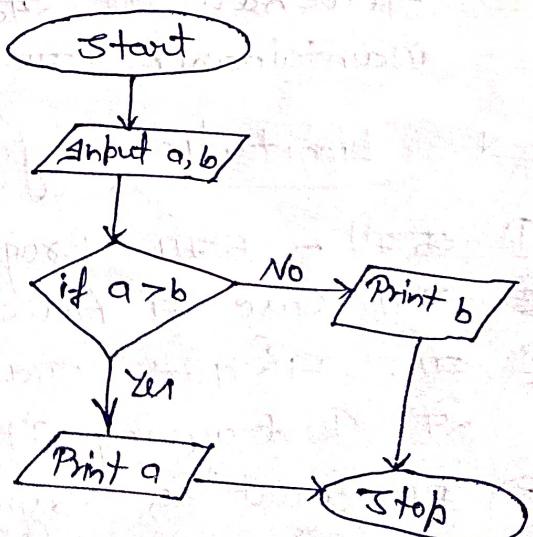
② DFD (Data Flow Diagram) of S.I. ?



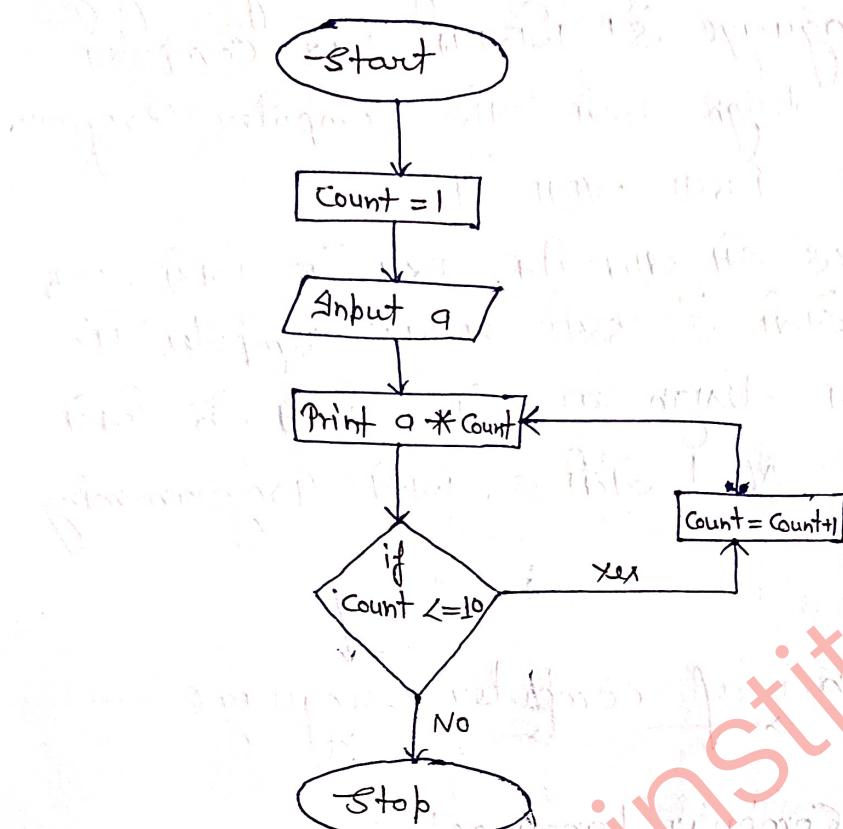
③ D.F.D. to check Even and Odd?



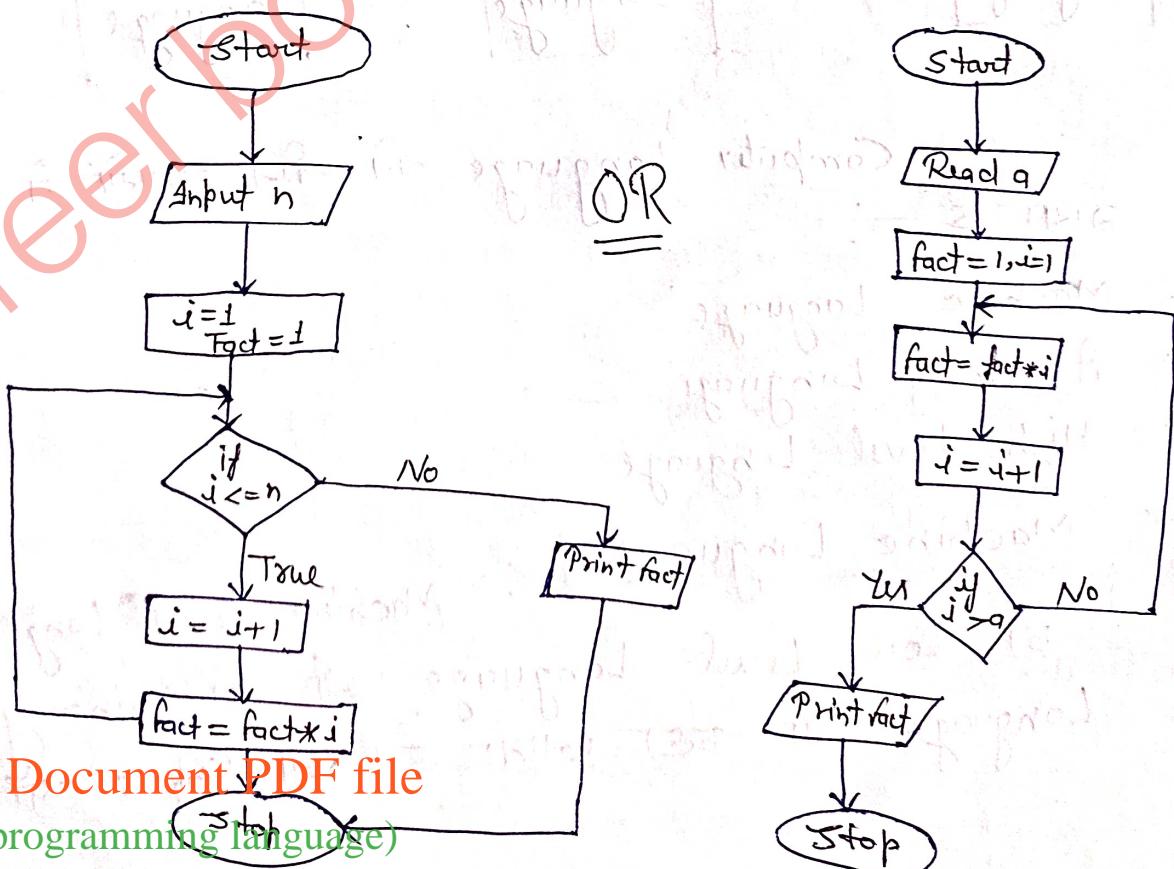
④ दो Value में से बड़ी Value को Find करने का D.F.D. बनाओ



Python programming language Number का Table Print करने के लिये D.F.D. बनाओ।



⑥ User द्वारा दिये गये Number का Factorial Find करने के लिये एक D.F.D. बनाओ।



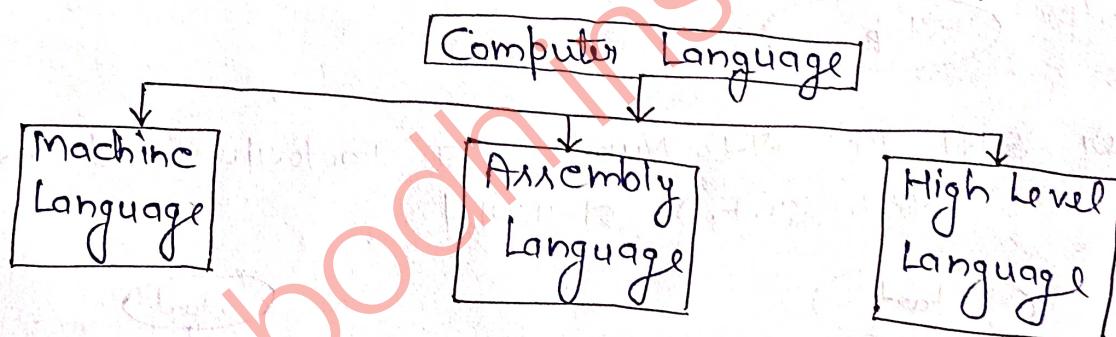
* Python programming language

Programming Language →

Computer Language है। जिसका Use Computer Program को design करने और Computer Program लिखने के लिये किया जाता है।

जिस तरह हमें बातचीत करने के लिये हमें Language की Need होती है उसी प्रकार Computer में किसी Program को design या कोड करने के लिये एक Language की Need होती है, जिसे Programming Language कहते हैं।

* Classification of Computer Language !



उत्तराः— Computer Language की तीन स्तरों में वाहः

1. Machine Language
2. Assembly Language
3. High Level Language

* Machine Language ↗ :

को Low Level Language कहा जाता है जो Binary level Language की कहा जाता है जो Input की

~~Python programming language~~ Output के रूप में केवल Binary Number की Information का Use करती है। Machine Level Language में Code बदलने के लिये किसी Mediator की Need नहीं होती। ताकि उसे Execution के लिये ज्ञान सके।

* Profit →

1. यह Computer का Fast तथा Efficient (कुशल) Use करता है।
2. इसके लिये किसी Translator की Need नहीं होती अमींकि यह सीधे Computer द्वारा समझा जाता है।

* Losses →

1. Binary और Hexa Code वर्गमूल में Unreadable होता है।
2. Machine Level Language व्याप्ति रखने तथा Debug करने के लिए कोड को लिखना है।
3. Machine Language में कोई Mathematical Function उपलब्ध नहीं होती।

* Assembly Level Language →

यह Language होती है, जो Machine Level Language की तुलना में इसकी विशेषताओं को बढ़ाती है। Assembly Language, Machine Level Language की तुलना में लिखने में थोड़ा fast or easy होती है। इसके लिये Assembler का Use होता है। जो Assembly Language की Low Level Language में convert करता है। Assembly Language

Mnemonics तथा Register Variable का Use करती है।

* Profit! → यह Machine Level Language की तुलना में थोड़ा fast है।

* Losses → इसमें Implement करना कठिन होता है। Implement के द्वारा Mnemonics प्राप्त रखना बहुत मुश्किल है।

Python programming language

* High Level Language → High Level Language
 यह Programming Language है। जिसे मानव आसानी से समझ सकता है और लिख सकता है। High Level Language का mean सामान्य English Language से ही जो मानव समझ के बहुत करीब है। High Level Programming Language का use करके Program Code को बहुत आसानी से और सही तरीके से लिखा जा सकता है।

Program Code को Binary Information में परिवर्तित करने के लिये Compiler तथा Interpreter का use करते हैं। कुछ High Level Language जैसे → C, C++, C#, Java, BASIC etc.

* Profit of High Level Language →

- ① High Level Language का use करने के कारण समझने और लागू करने में आसान है।
- ② Machine Level तथा Assembly Language की तुलना में यह तेज होती है।
- ③ यह मशीन पर नियंत्रित होती है। इसको हम किसी Machine पर चला सकते हैं।

* Losses! →

Integrated development environments

- ① Implementation के दौरान Application (IDE) की समझने के लिये आवश्यक skill जी जी Need होती है।
- ② Application के All components की वजह से लिये Memory रूप से आकर्षण है।

Python programming language

ALGOL → Algorithmic Language

BASIC → Beginner's All purpose Symbolic Instruction Code

COBOL → Common Business Oriented Language

FORTRAN → Formula Translator

LISP → List Processor

*. Linker

Source program

Assembler

Object
Code

Linker

Executive
Code
↓
Loader

Linker एक System Software है, जो Binary Language में प्राप्त Code की मालिन पर अपलने लायक मालिन Code में बदल देता है।

जो Computer Language में Language है उसे Source Language कहते हैं; या हम उसे Source Code के नाम से भी जानते हैं। और इस Source Code को जिस Language में बदलाव जाता है। उसे Target Language कहते हैं। इससे प्राप्त Code को Object Code कहते हैं। Object Code की Linker, Machine Code में बदलता है Application Extension .exe प्रकार की File की Linker की Develop करता है।

Python programming language

* Loader → Loader O.S. का एक हिस्सा है, जो Disc पर Memory में रहने वाली एक Executable फ़ॉरम फ़ाइल लाता है और उसे Start करता है। Linker के बाद आगे का काम Loader पर ही होता है। Linker के द्वारा Generate किये गये Machine Code को memory में Load करने का काम Loader का ही होता है।

Loader Program के virtual address की Physical address में बदल देता है जिसके परिणामस्वरूप file तथा folder भेयार ही जाते हैं। दोनों में अंतर यह है कि linker program execution फ़ॉरम नहाता है जबकि Loader Execution के लिये Linker से प्राप्त files की Main memory में Load करता है।

* Testing → Testing Programming में किसी Website पर software का जीतना step होता है जिसके द्वारा हम अपने Project की All functionality को check करते हैं। Testing का use program के result से सम्बन्धित सभी things की सुनिश्चित करने के लिये किया जाता है। goal की ढीक करने के लिये जो भी आवश्यक लगता है उसके according हम उसे ढीक कर सकें। किसी Project की वजह से testing की बहुत आवश्यक Need होती है। इस प्रयोग की वजह की सही से check कर सकते हैं, किसी वीज में कोई Problem आती है तो उसका हम पता चल जाता है और हम उसकी ही कर

~~Testing~~ विश्वासन प्रकार की होती है।

1. Functional Testing : → इस प्रकार की Testing के द्वारा किसी Project की All functionality check की जाती है।
2. Performance Testing : → इस Testing का Use उसकी Speed की check करने के लिये किया जाता है।
3. Stress Testing : → ऐसे पड़ Testing होती है जिसके द्वारा check करने के लिये किया जाता है कि System प्रतिकूल परिस्थितियों में कैसा व्यवहार करता है।
4. Integration Testing : → Integration Testing वह Testing है। जिसमें Component के एक ग्रुप को उत्पादन के लिये संयुक्त किया जाता है। इसके अलावा Hardware तथा Software के बीच After action का Test Integration Testing के द्वारा किया जाता है।
- *. Debugging : → Debugging वह Process है, जिसमें Program में Error आती है, और उन्हें ढीकु किया जाता है। दूसरे शब्दों में हम कह सकते हैं कि किसी Program को Compile करते समय कोई त्रुटि दिखाई देती है और उसको ढीकु करने की Process को Debugging कहा जाता है। इसमें विश्वासन प्रकार की Error आती है।

Python programming language

* Syntax Error → Syntax Error गलत Syntax लिखने से Generated होती है। इसके लिए grammatical error मी कहते हैं।

for Example → जब हम किसी Function की define करते समय (:) लगाना मूल जाते हैं। इस प्रकार की Error की Syntax Error कहते हैं। ऐसे- हम कोई Program लिखते समय if Condition का use करते हैं; तब if $m > 0$ लिखते हैं। और इसके बाद में colun(:) लगाना मूल जाते हैं तो इस Error की Syntax Error कहते हैं।

* Runtime Error → सभी Syntax Error और Semantic Error को Compiler द्वारा पता लगाया जाता है। जो इस संदेश उत्पन्न करता है। जिसमें Error का प्रकार और source file में स्थित जहाँ भूट है। लाइनिंग कहते हैं।

Runtime Error के Error होते हैं। जो Code Execution के दौरान आते हैं। ये लड़त-लानिकारक होती है। क्योंकि चलते Program की ये रोक देती है।

* Semantic Error → Program Statement के Improper Use के कारण जो Error आती है, उसको Semantic Error कहते हैं।

* Logical Error → Logical Error इस Error होती है। जो इस Program के source code में जिससे कि output में error या unexpected change दिखाई देते हैं, ये Error वह आती है जब Software Specification का सम्मान नहीं किया जाता है।

Python programming language

* Testing & Debugging में अन्तर -

Testing

- ① Testing हमेशा Known Condition के साथ शुरू होता है।
- ② यह Programmer को failure साक्षित करता है।
- ③ यह एप्पल मा स्पष्ट शुद्धता का प्रदर्शन है।
- ④ Testing किसी outsider का नियंत्रण की जाती है।
- ⑤ वह सारे test बिना design ज्ञान के किये जाते हैं।
- ⑥ Testing का Purpose bug को find करना है।
- ⑦ Testing certainly नियोजित, design और निर्धारित किया जाना चाहिए।

Debugging

- ① Debugging Unknown Condition के साथ शुरू होता है।
- ② यह Programmer को स्क्रॉट करता है।
- ③ यह हमेशा एक deductive program के रूप में साना जाता है।
- ④ Debugging इमिडियट कारा की जाती है।
- ⑤ इसमें design knowledge का हीना छहत आवश्यक है।
- ⑥ Debugging का Purpose bug का कारण ढूँढ़ना है।
- ⑦ Debugging की जबाबदी और अवाधि की Project इतनी constant नहीं हो सकती।

* Documentation

सभी कड़ी Software development project पाते ही जो भी हो, कड़ी मात्र में सम्बन्ध Document बनाते हैं। ऐसे- files की एक अलगावी में रखना और आदि कड़ी files हो तो उन्हें कमरे में भर देते हैं। इस Document के निमित्त में Software Process लागत का एक उच्च अनुपात खाये होते हैं। इसके अलावा Documentation के छारा हम किसी भी document को आसानी से प्राप्त कर सकते हैं।

Document की प्रकार के होते हैं:-

- ① User document
- ② Technical document

Python Document PDF file

(Python programming language)

Python programming language

1. User Document → User document एक Product का वास्तविक service के लिये documentation को Refer करता है। जो अंतिम User को प्रदान किया जाता है। User documentation Product का वास्तविक service का UIC करने के लिये अंतिम User की सहायता के लिये design किया जया है। इसे अक्सर User सहायता के रूप में जाना जाता है।

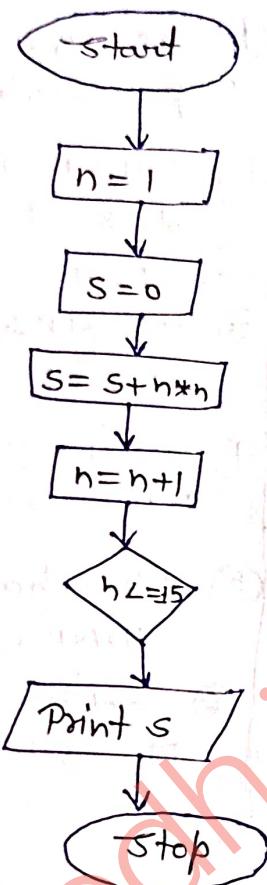
2. Technical Document / System Document → Engineering में Technical Documentation किसी ऐसी प्रकार के Documentation में एक Technical Product का UIC के लिये Product handling और functionality को describle करता है। Technical documents का purpose इसमें Product की Inner और Outer विवरण की समझने के लिये User के लिये प्राप्त जानकारी प्रदान करता है।

- * Aim for Documentation →
- ① Documentation के द्वारा छोखाई और दुर्घटनाएँ को कम किया जा सकता है।
 - ② ग्राहकों की Needs का अनुपालन
 - ③ Union and State & Local Regulation का अनुपालन
 - ④ Trained New Employees

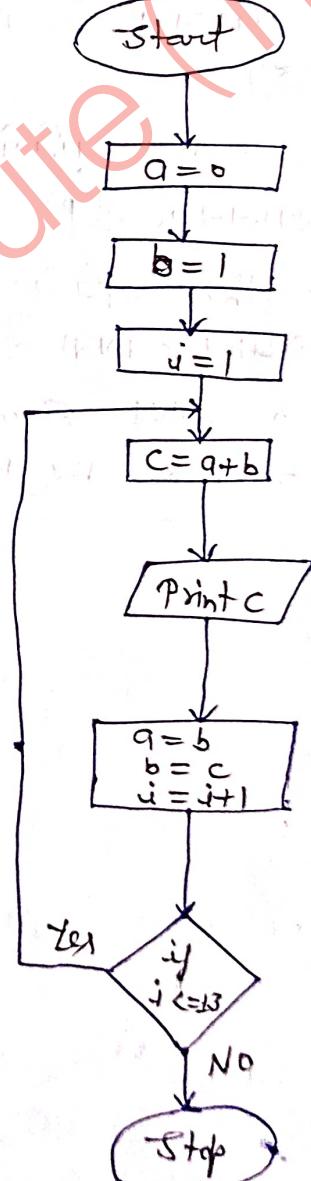
Python programming language

* Draw a flowchart to calculate and print sum of the following series:-

1, 4, 9, 16, 25, ... up to 15 term.



Q. Fibonacci Series का D.F.D. बनाओ
0, 1, 1, 2, 3, 5, 8, 13, ...



Python programming language

* Difference between Algorithm and Flow chart :

Algorithm

- ① यह Problem Solve करने का एक Procedure है।
- ② इस Process को step by step Instruction में प्राप्तार्थीत किया जाता है।
यह समझने के लिये difficult होता है।
- ③ यह debug 'प्रूटियो' के लिये सुविधाजनक है।
- ④ Algorithm बनाने में अधिक समय लगता है।
- ⑤ इसके द्वारा Complex Problem को Solve करना आसान होता है।

Flow chart

- ① यह किसी Process का Graphic Representation है।
- ② इसकी Process को block by block Information diagram में दिखाया जाता है।
- ③ यह समझने के लिये आसान होता है।
- ④ इसमें 'प्रूटियो' को Debug करना कठिन है।
- ⑤ Flow chart बनाने में कम समय लगता है।
- ⑥ इसके द्वारा Complex Problem की Solve करना कठिन होता है।