Unit 3

Principles of Programming and Problem solving :

കമ്പ്യൂട്ടറിന് മനസ്സിലാകുന്ന ഭാഷയിൽ നൽകുന്ന , പ്രശ്നപരിഹാരത്തിന് ആവശ്യമായ ഒരു കൂട്ടം നിർദ്ദേശങ്ങളാണ് computer program. കമ്പ്യൂട്ടറിന് സാമാന്യ ബുദ്ധി ഇല്ലാത്തതിനാൽ, പ്രശ്നപരിഹാരത്തിന്റെ യുക്തിക്കനുസരിച്ച് , നിർദ്ദേശങ്ങൾ വ്യക്തമായും ക്രമമായും പ്രോഗ്രാമർ എഴുതണം.

Approaches in Problem solving: പ്രശ്നപരിഹാരത്തിന് രണ്ടുതരം സമീപനങ്ങൾ ഉണ്ട്.

1. Top down design. വലിയ പ്രോഗ്രാമുകളെ ചെറിയ സ്വതന്ത്രമായ പ്രോഗ്രാമുകളാക്കി (Modules) മാറ്റുന്നു. മുഖ്യ പ്രശ്നത്തിന്റെ മേൽത്തട്ടുമുതൽ ആരംഭിക്കുകയും അതിലെ ഓരോ ഉപവിഭാഗത്തിനും പരിഹാരങ്ങൾ കാണുന്ന ഈ രൂപകൽപ്പനയിൽ. പ്രോഗ്രാമിനെപ്പറ്റി വ്യക്തമായ ധാരണയുണ്ടാകുക വഴി പ്രശ്നപരിഹാരം എളുപ്പമാകും. ഒന്നിലധികം ആളുകൾക്ക് പ്രശ്നപരിഹാരത്തിൽ പങ്കാളികളാകാം. Module കൾ പുനരുപയോഗിക്കാം. (ഉദാ. വീടിന്റെ പ്ലാൻ വരക്കുന്നത് ഓർക്കുക) .
2. Bottom up design. ഇവിടെയും വലിയ പ്രോഗ്രാമുകളെ ചെറിയ പ്രോഗ്രാമുകളാക്കുകയും വീണ്ടും ചെറിയ മൊഡ്യൂളുകളാക്കുകയും ഇവയ്ക്ക് പരിഹാരം കണ്ട് എല്ലാം കൂട്ടിച്ചേർത്ത് പ്രശ്നപരിഹാരം കാണുന്ന രീതിയാണിത്. (ഉദാ. വീടിന്റെ നിർമ്മാണപ്രവർത്തനം ഓർക്കുക)

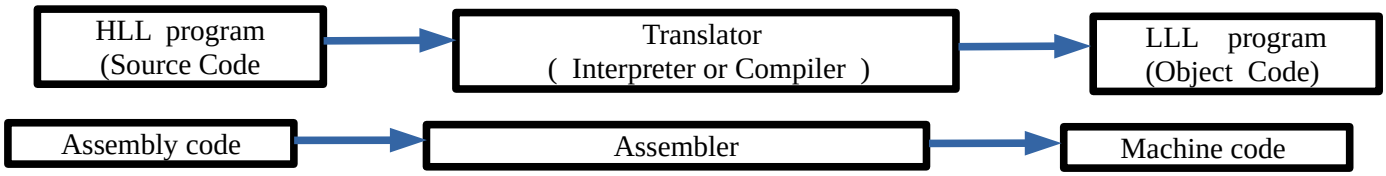
Phases in programming (**പ്രോഗ്രാമിന്റെ ഘട്ടങ്ങൾ**) :

1. Problem Identification : പ്രശ്നത്തെപ്പറ്റി വ്യക്തമായി മനസ്സിലാക്കൽ. ആവശ്യമായ ഡാറ്റകളും അവയുടെ തരവും, അവയിലുൾപ്പെട്ട പ്രവർത്തനങ്ങൾ, അന്തിമ ഫലം തുടങ്ങിയവ.

2). Preparing of Algorithms and flowchart : പ്രശ്ന പരിഹാരത്തിനുള്ള ക്രമമായ നിർദ്ദേശങ്ങളുടെ കൂട്ടമാണ് Algorithm.(ഒരു പലഹാരം ഉണ്ടാക്കുന്നതിന്റെ recipe ഓർക്കുക). അൽഗോരിതത്തിന്റെ ചിത്രീകരണമാണ് flowchart. ജ്യാമിതീയ രൂപങ്ങൾ ഉപയോഗിച്ച് വരക്കുന്ന ചിത്രീകരണമായതിനാൽ , പ്രവർത്തനക്രമവും യുക്തിയുമൊക്കെ അൽഗോരിതത്തേക്കാൾ എളുപ്പത്തിൽ മനസ്സിലാക്കാം. [ഈ ചിഹ്നങ്ങൾ ആഗോളതലത്തിൽ ANSI (American National Standard Institue) യുടെ അംഗീകാരത്തോടെ ഉപയോഗിക്കുന്നതാണ്.]

3). Coding : കമ്പ്യൂട്ടർ ഭാഷയിൽ program code എഴുതൽ. ഇങ്ങനെ മനുഷ്യർക്ക് മനസ്സിലാകുന്ന ഭാഷയിൽ (High Level Language) എഴുതുന്ന കോഡാണ് source code. കോഡ് തയ്യാറാക്കാൻ വേണ്ടി ഉപയോഗിക്കുന്ന ഓരോ പ്രോഗ്രാമിംഗ് ഭാഷയ്ക്കും അതിന്റേതായ അക്ഷരമാലകളും (character set), പദാവലികളും (vocabulary), വാക്യഘടനയും (syntax) ഒക്കെ ഉണ്ടാകും. Eg. BASIC, C, C++, Java, Python തുടങ്ങിയവയൊക്കെ വ്യത്യസ്ത പ്രോഗ്രാമിംഗ് ഭാഷകളാണ്.

4). Translation : HLL ൽ എഴുതിയ code , machine code ലേക്ക് (LLL - Low Level Language) മാറ്റുന്ന പ്രക്രിയ. ഇതിന് Compiler or interpreter ഉപയോഗിക്കുന്നു.



5). Debugging : program code ലുണ്ടാകുന്ന തെറ്റുകൾ (bugs) കണ്ടെത്തുകയും പരിഹരിക്കുകയും ചെയ്യുന്നത്.

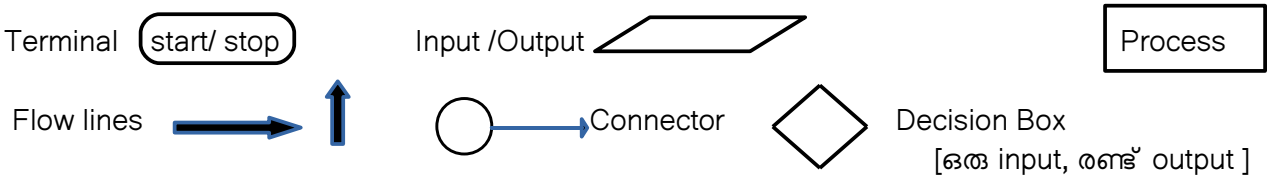
6). Execution and testing : program execute ചെയ്യുന്നു. ഉചിതമായ, വിവിധ data കൾ നൽകി കിട്ടുന്ന ഫലങ്ങൾ ശരിയാണോ എന്നും പരിശോധിക്കുന്നു. കമ്പ്യൂട്ടർ ഉപയോഗിക്കാതെതന്നെ നാം ഇത്തരം ഡാറ്റകൾ നൽകി ക്രിയകൾ ചെയ്ത് കിട്ടുന്ന ഫലവും, കമ്പ്യൂട്ടറിൽ ചെയ്ത് കിട്ടുന്ന ഫലവും താരതമ്യം ചെയ്യേണ്ടിവരും.

7). Documentation : പ്രോഗ്രാമിനെക്കുറിച്ചും ഉപയോഗരീതിയെക്കുറിച്ചും വിശദീകരിക്കുന്ന കുറിപ്പുകൾ. പ്രോഗ്രാമിൽ ലഘു വിവരണങ്ങൾ നൽകുന്നതുവഴി മറ്റൊരാൾക്കോ, പിന്നീട് നമുക്കോ , പ്രോഗ്രാമിന്റെ യുക്തിയെക്കുറിച്ച് മനസ്സിലാക്കാനും പ്രോഗ്രാമിൽ മാറ്റം വരുത്താനും എളുപ്പത്തിൽ കഴിയും. പ്രോഗ്രാമിന്റെ ഭാഗമായി എഴുതുന്ന ഇത്തരം വിവരണങ്ങൾ compilation സമയത്ത് പരിഗണിക്കില്ല.

Documentation രണ്ട് വിധം ഉണ്ട്. . 1. Internal Documentation: source code ൽ comment എഴുതുന്നതാണിത്. 2. Externall Documentation: user manual ഉം system manual ഉം തയ്യാറാക്കുന്നതാണിത്. പ്രോഗ്രാമുകളുടെ ആവശ്യകത, പ്രവർത്തനം, ഉപയോഗരീതി തുടങ്ങിയ വിവരണങ്ങളാണ് ഇതിലുണ്ടാവുക.

Hard copy - പേപ്പറിൽ ലഭിക്കുന്ന വിവരങ്ങൾ, ചിത്രങ്ങൾ , ഡാറ്റകൾ തുടങ്ങിയവയുടെ പ്രിന്റൗട്ടാണ് ഉദ്ദേശിക്കുന്നത്. **Soft copy** : Electronic ഉപകരണങ്ങൾ വഴി മാത്രം കാണാൻ കഴിയുന്ന വിവരങ്ങൾ, ചിത്രങ്ങൾ , ഡാറ്റകൾ തുടങ്ങിയവ.

Flow chart ൽ ഉപയോഗിക്കുന്ന ചിഹ്നങ്ങൾ .



Algorithm: സവിശേഷതകൾ . Start statement ൽ തുടങ്ങി, ആവശ്യമായ ഇൻപുട്ടുകൾ സ്വീകരിച്ചുകൊണ്ട്, ചുരുങ്ങിയതും വ്യക്തവുമായ , നടപ്പിലാക്കാൻ പറ്റുന്ന നിർദ്ദേശങ്ങൾ ക്രമമായി എഴുതി stop statement ൽ അവസാനിക്കണം. നിർദ്ദേശങ്ങൾ ആവർത്തിക്കുന്നു എങ്കിൽ അത് പരിമിതമാവണം. അൽഗോരിതത്തിന്റെ അവസാനം പ്രതീക്ഷിക്കുന്ന ഫലം കിട്ടണം. ഒരു പ്രശ്നപരിഹാരത്തിന് വ്യത്യസ്ത അൽഗോരിതങ്ങൾ എഴുതാം. ചുരുങ്ങിയ നിർദ്ദേശങ്ങൾ കൊണ്ട് , കുറഞ്ഞ മെമ്മറി ഉപയോഗിച്ച് , വേഗത്തിൽ ഫലം കിട്ടുന്ന അൽ ഗോരിതമാണ് Best Algorithm.

Flowcharts: ചിത്രീകരണമായതുകൊണ്ട് Algorithm ത്തേക്കാൾ മെച്ചം Flowcharts ആണ്. ***സവിശേഷതകൾ .*** പ്രശ്നത്തെ എളുപ്പത്തിൽ, യുക്തിപരമായി വിശകലനം ചെയ്യാൻ കഴിയും. സങ്കീർണ്ണമായ പ്രവർത്തനഘട്ടങ്ങളെ ലഘൂകരിക്കാം. സങ്കീർണ്ണമായ പ്രോഗ്രാമുകളെ മൊത്തത്തിൽ വിശകലനം ചെയ്യാം. പ്രോഗ്രാമിൽ വരാൻ സാധ്യതയുള്ള തെറ്റുകൾ കണ്ടെത്താനും തിരുത്താനും കഴിയും. നല്ല പ്രോഗ്രാം കോഡ് എഴുതാൻ ഇത് സഹായകരമാണ്. ***പരിമിതികൾ***. വരക്കാൻ കൂടുതൽ സമയം വേണം. വരച്ചതിൽ മാറ്റം വരുത്തണമെങ്കിൽ പുതുതായി മറ്റൊന്ന് വരക്കണം. വരക്കുന്നതിന് നിശ്ചിത നിയമാവലികൾ (standards) ഇല്ല.

Debugging: Errors 3 വിധമുണ്ട്. **1. Syntax error**: program ഭാഷയിലെ നിയമങ്ങൾ തെറ്റിയാൽ വരുന്ന error. Compile സമയത്ത് വരുന്ന error ആണിത്. Eg. Punctuation mark തെറ്റായി നൽകുന്നത്. **2. Logical error**. തെറ്റായ out put കിട്ടും. programming എഴുതുന്നതിലെ യുക്തിക്കുറവുകൊണ്ടാണ് ഇത്തരം തെറ്റുകൾ വരുന്നത്. Compile സമയത്ത് error കാണില്ല. 3. **Run time error** : Program run ചെയ്യുമ്പോൾ തെറ്റായ input data നൽകിയാൽ വരുന്ന error . Eg. A / B എന്ന ക്രിയയിൽ B = 0 ആയാൽ ഈ error വരും.

Loops: പ്രശ്നപരിഹാരത്തിന്റെ ഭാഗമായി ചില പ്രവർത്തനങ്ങൾ ആവർത്തിക്കേണ്ടതായി വരും. ഇത്തരം സന്ദർ ഭങ്ങളിൽ ലൂപ്പിന്റെ ആവശ്യം വരും. ഒരു ലൂപ്പിന് നാല് ഘടകങ്ങൾ ഉണ്ട്. ഒരു **നിബന്ധന (condotion)** ശരിയാകുന്നതുവരെ സാധാരണയായി പ്രവർത്തനം തുടരേണ്ടിവരും. വ്യവസ്ഥ നൽകുന്നതിന് വേണ്ടി ഒരു വേരിയബിളെങ്കിലും (ഈ വേരിയബിളാണ് ലൂപ്പ് കൺട്രോൾ വേരിയബിൾ) വേണം. ഈ വേരിയബിളിന് തുടക്കവില നൽകാനുള്ള നിർദ്ദേശങ്ങൾ (**Initialisation Instruction**) നൽകണം. കൂടാതെ loop control variable ന് മാറ്റങ്ങൾ വരുത്തേണ്ട നിർദ്ദേശങ്ങളും (updation instruction) നൽകേണ്ടിവരും . ആവർത്തിച്ച് പ്രവർ ത്തിക്കേണ്ട നിർദ്ദേശങ്ങൾക്ക് **Body of the loop** എന്ന് പറയുന്നു.

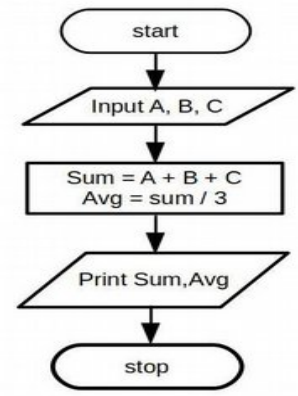
നിബന്ധനകൾ ശരിയാണെങ്കിൽ മാത്രം ലൂപ്പ് ബോഡി പ്രവർത്തിക്കുന്ന തരം ലൂപ്പുകൾ ഉണ്ട്. ഇവ **Entry Controlled Loop** എന്ന പേരിൽ അറിയപ്പെടുന്നു. ഇത്തരം ലൂപ്പുകളിൽ നിബന്ധന ശരിയല്ലെങ്കിൽ ലൂപ്പ് ബോഡി ഒരിക്കലും പ്രവർത്തിക്കില്ല. എന്നാൽ നിബന്ധനകൾ ശരിയായാലും തെറ്റായാലും ലൂപ്പ് ബോഡി ചുരുങ്ങിയത് ഒരു തവണ പ്രവർത്തിക്കുന്ന തരം ലൂപ്പുകളും ഉണ്ട്. ഇവ **Exit Controlled Loop** എന്ന പേരിൽ അറിയപ്പെടുന്നു. തുടർ ന്ന് ലൂപ്പ് ബോഡി പ്രവർത്തനക്ഷമമാവണമെങ്കിൽ നിബന്ധന ശരിയാവണം.

Algorithm and Flow chart. Some Examples.

Write Algorithm and draw Flowchart to find the sum and average of three numbers.

Algorithm:

1. Start
2. Input A, B, C
3. Sum = A + B + C
4. AVG = Sum /3
5. Print Sum, AVG
6. Stop

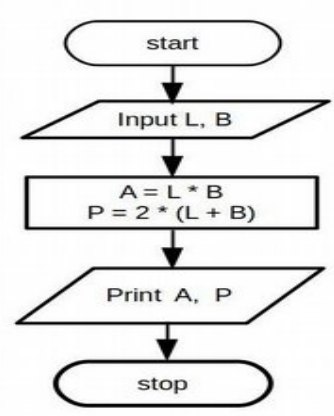


Write Algorithm and draw Flowchart to find the area and perimeter of a rectangle.

(Area = length * breadth,
perimeter = 2* (Length + Breadth)

Algorithm:

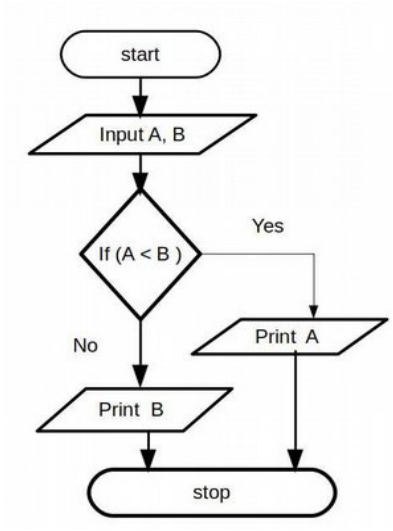
1. Start
2. Input L, B
3. A = L * B
4. P= 2 (L + B)
5. Print A, P
6. Stop



Write Algorithm and draw Flowchart to find the smaller number from the two numbers.

Algorithm:

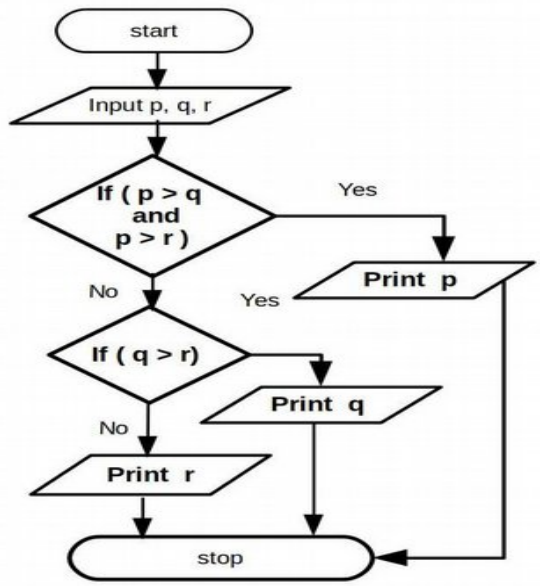
1. Start
2. Input A. B
3. if (A < B) then
4. Print A
5. else
6. print B
7. Stop



Write Algorithm and draw Flowchart to find the biggest number from the three numbers .

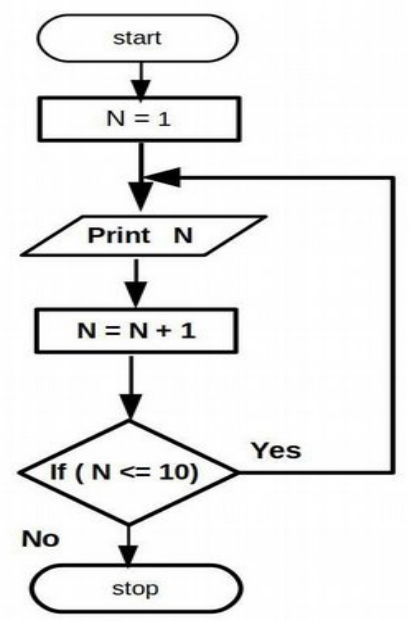
Algorithm:

1. Start
2. Input p,q, r
3. if (p > q and p > r) then
4. Print p
5. else if (q > r) then
6. print q
7. else
8. print r
9. stop



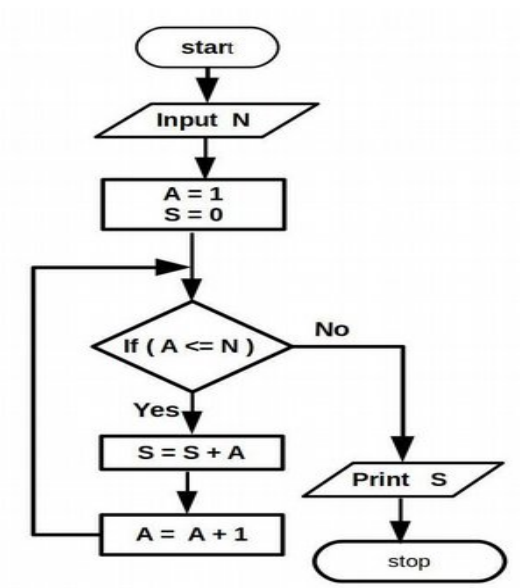
Write Algorithm and draw Flowchart
to print numbers from 1 to 10 .

Algorithm:
1. Start
2. N = 1
3 print N
4. N = N + 1
5. if (N <= 10) then go to step 3
6. stop



Write Algorithm and draw Flowchart
to print sum of first N natural Numbers

Algorithm:
1. Start
2. Input N
3 A =1, S = 0
4. Repeat step 5 and 6 while (A <=N)
5. S = S + A
6. A = A + 1
7. Print S
8. stop



Unit 3. Principles of Programming & Problem Solving

1. In a flowchart, the terminal (an ellipse) symbol is used to indicate the ..... and ..... in the program logic.
2. Pictorial representation of an algorithm is known as .....
3. The process of converting source code into object code is called ..... (source code നെ object code ലേക്ക് മാറ്റുന്ന പ്രക്രിയയെ ..... എന്ന് വിളിക്കുന്നു)
4. ..... is a step by step procedure to solve a problem. (ഒരു പ്രശ്നം പരിഹരിക്കുന്നതിന് വേണ്ടിയുള്ള ഘട്ടം ഘട്ടമായുള്ള നടപടിയാണ് .....) (1)
5. Which one of the following is NOT a part of program documentation. (താഴെ കൊടുത്തവയിൽ program documentation ന്റെ ഭാഗം അല്ലാത്തത് ഏത് ? (1)
   a). Writing comments in the source code. (source code ൽ വിവരണങ്ങൾ എഴുതുന്നത്.)
   b). Detecting and correcting errors. (തെറ്റുകൾ കണ്ടെത്തി തിരുത്തുന്നത്)
   c). Preparation of system manual. (system manual തയ്യാറാക്കൽ)
   d). Preparation of user manual. (user manual തയ്യാറാക്കൽ)
6. Write short notes on (ചെറു വിവരണം തയ്യാറാക്കുക) a) Coding b) Debugging (2)
7. Explain any two limitations of flow chart. (ഫ്ലോചാർട്ടിന്റെ ഏതെങ്കിലും രണ്ട് പോരായ്മകൾ എഴുതുക) (2)
8. Draw a flowchart to input three numbers and display the smallest. (മൂന്ന് സംഖ്യകൾ ഇൻപുട്ട് ചെയ്ത് അവയിൽ ചെറുതിനെ കണ്ടുപിടിക്കാനുള്ള ഫ്ലോചാർട്ട് വരയ്ക്കുക) (3)
9. Problem solving by computer proceeds through different stages. Name the stages in correct order . (കമ്പ്യൂട്ടർ ഉപയോഗിച്ചുകൊണ്ടുള്ള പ്രശ്നപരിഹാരം പല ഘട്ടങ്ങളിലൂടെയാണ് കടന്ന് പോകുന്നത്. പ്രസ്തുത ഘട്ടങ്ങളെ ക്രമത്തിലെഴുതുക) (2)

10. List the different stages in programming (പ്രോഗ്രാമിംഗിലെ വിവിധ ഘട്ടങ്ങൾ എഴുതുക) (3)

11. Write any two limitations of flowchart (ഫ്ലോചാർട്ടിന്റെ ഏതെങ്കിലും രണ്ട് പരിമിതികൾ എഴുതുക)

12. Define the following (a). Syntax error (b). Logical error (c) Runtime error (3)

13. Draw the flowchart for the algorithm given below (algorithm ത്തിന് അനുസൃതമായ flowchart വരക്കുക) (3)

Step 1 : Start
Step 2 : Input A, B
Step 3 : If A > B Then
Step 4 : Print A
Step 5 : Else
Step 6 : Print B
Step 7 : End of If
Step 8 : Stop

14. Write an algorithm to print first 100 natural numbers. (3)

15 . a). List different phases in programming. (പ്രോഗ്രാമിംഗിന്റെ വിവിധ ഘട്ടങ്ങളുടെ പേരെഴുതുക)

b) Explain any three phases in programming. (ഏതെങ്കിലും മൂന്ന് പ്രോഗ്രാമിന്റെ ഘട്ടങ്ങളെക്കുറിച്ച് വിവരിക്കുക) (5)

16. Write an algorithm to find the biggest of two numbers (രണ്ട് സംഖ്യകളിൽ നിന്ന് വലിയ സംഖ്യ കണ്ടുപിടിക്കുന്നതിനുള്ള അൽഗോരിതം എഴുതുക) (3)

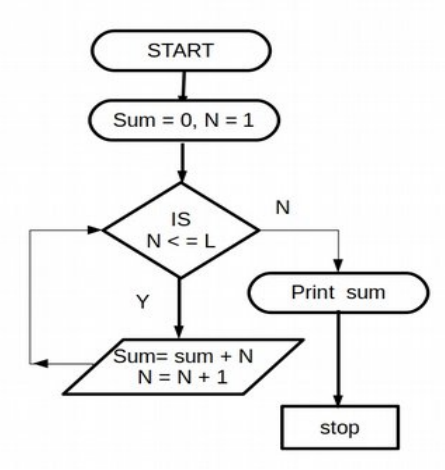
17. Draw a flowchart to find the sum and average of three given numbers. (തന്നിരിക്കുന്ന മൂന്ന് സംഖ്യകളുടെ തുകയും ശരാശരിയും കാണുവാനുള്ള flowchart വരയ്ക്കുക) (3)

18. Explain two types of documentation in programming . (programming ലെ രണ്ട് തരം ഡോക്യുമെന്റേഷനുകൾ വിവരിക്കുക) (3)

19

a) correct the flowchart if there are errors.
(ഫ്ലോചാർട്ടിൽ തെറ്റുകളുണ്ടെങ്കിൽ തിരുത്തുക) (2)

b) Redraw the flowchart to find the sum of even numbers between 1 and 50. (1 നും 50 നും ഇടയിലുള്ള സംഖ്യകളുടെ തുക കാണുന്നതിനുള്ള ഫ്ലോചാർട്ടാക്കി ഇതിനെ മാറ്റി വരക്കുക) (1)



20 Write an algorithm to print the multiples of 5 between 100 and 200 in descending order.
(100 നും 200 നും ഇടയിലുള്ള 5 ന്റെ ഗുണിതങ്ങളെ അവരോഹണ ക്രമത്തിൽ കാണുന്നതിനുള്ള അൽഗോരിമെഴുതുക) (3)

21. Errors may occur in two stages of programming . (പ്രോഗ്രാമിംഗിന്റെ രണ്ട് ഘട്ടങ്ങളിൽ തെറ്റുകൾ സംഭവിക്കാം.)

a). Name these two stages. Explain the nature of errors in these stages . (ഈ ഘട്ടങ്ങളുടെ പേരെഴുതി തെറ്റുകളുടെ സ്വഭാവം വിശദീകരിക്കുക) (3)

b). The process of correcting these errors is known as . ..... (ഈ തെറ്റുകൾ ശരിയാക്കുന്ന പ്രക്രിയ ..... എന്നറിയപ്പെടുന്നു) (1)

22. Draw any six flowchart symbols and specify their standardised meaning . (ഏതെങ്കിലും ആറ് flowchart symbols വരക്കുകയും അവയുടെ വ്യവസ്ഥാപിത അർത്ഥം വ്യക്തമാക്കുകയും ചെയ്യുക) (3)

23. Explain the different types of errors that may occur in a program .(ഒരു പ്രോഗ്രാമിൽ സംഭവിച്ചേക്കാവുന്ന വിവിധതരം തെറ്റുകളെക്കുറിച്ച് വി Write an algorithm to print first 100 natural numbers. (3)ശദീകരിക്കുക. (3)

24. An algorithm is a finite sequence of instruction to solve a problem. (Algorithm എന്നാൽ ഒരു പ്രശ്നത്തിന്റെ ഉത്തരം കണ്ടെത്താനുള്ള നിശ്ചിതമായതും തുടർച്ചയായതുമായ നിർദ്ദേശങ്ങളാണ്.)

a) What are the characteristics of algorithm . (Algorithm ത്തിന്റെ സവിശേഷതകൾ എന്തൊക്കെ) (3)

b). Pictorial representation of algorithms is called ..... (Algorithm ത്തെ ചിത്ര രൂപത്തിൽ കാണിക്കുന്നതിനെ ..... എന്ന് വിളിക്കുന്നു.) (1)

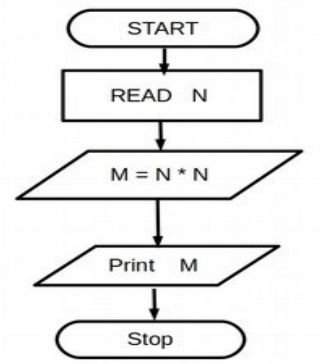
25. Draw a flowchart to find the sum and average of even numbers between 1 and 99. (1 ന്റെയും 99 ന്റെയും ഇടയിലുള്ള എല്ലാ ഇരട്ട സംഖ്യകളുടേയും തുകയും ശരാശരിയും കാണുന്നതിനുള്ള ഒരു flowchart വരക്കുക) (3)

26. Write a short note on the importance of internal documentation. (internal documentation ന്റെ പ്രാധാന്യത്തെപ്പറ്റി ലഘുവായി വിവരിക്കുക) (3)

27. Write an algorithm to accept two numbers and display the biggest among them. (രണ്ട് സംഖ്യകളിൽ വലുത് കണ്ടെത്താനുള്ള അൽഗോരിതം എഴുതുക) (3)

28. Explain Debugging.

29. Redraw the following flowchart by correcting mistakes.

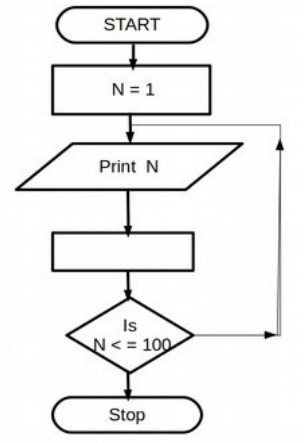


30. Consider the following algorithm :
Step 1 : Start
Step 2 : N = 1
Step 3 : Print N
Step 4 : N = N + 1
Step 5 : If N < = 5 then goto step 3
Step 6 : stop
i) . Write the output of the above algorithm (2)
ii). Draw the flow chart of the above algorithm (3)

31. Define the following three terms in programming (3 + 2)
a). Translation b) Debugging c) . Documentation
d). What are the criteria to choose the best algorithm for a problem . (ഏതെല്ലാം മാനദണ്ഡങ്ങൾ ഉപയോഗിച്ചാണ് ഒരു നല്ല അൽഗോരിതം തെരഞ്ഞെടുക്കുന്നത്.)

32. There are two popular approaches in problem solving. Describe each. (problem solving ന് സാധാരണയായി രണ്ട് സമീപനങ്ങളുണ്ട്. ഓരോന്നും വിശദീകരിക്കുക) (3)

33. The process of detecting and correcting errors in a program is called .....
34 Write an algorithm to find the sum of numbers from I to 10.
35. Explain any two types of programming errors.
36. ..... is the process of converting a program written in HLL into its equivalent machine language.
37. Flowchart to print numbers from 1 to 100 is given here.



a). Fill the missing element . (1)
b). Redraw the flowchart to print from 100 to 1 (2)

38. "The program written by one person may need to be modified by some other Person in future" (ഒരാൾ എഴുതിയ പ്രോഗ്രാം മറ്റൊരാൾക്ക് ഭാവിയിൽ മെച്ചപ്പെടുത്തേണ്ടി വന്നേക്കാം.

a). Which Phase of Programming will help for this ? (Programming ന്റെ ഏത് Phase ആണ് ഇതിന് സഹായിക്കുന്നത്.) (1)

b). Explain different methods used in the above phase (മുകളിൽ പറഞ്ഞിരിക്കുന്ന Phase ൽ ഉപയോഗിക്കുന്ന വിവിധ രീതികൾ വിവരിക്കുക) (2)