

Basics About Computer Language

→ Computer hardware only understands 0 & 1's i.e., binary system of numbers as commands.

Computer only understand binary language because.

- i) Reliable way for data storing
- ii) Since, the computer memory consists of many transistors, 0 & 1 is used to switch betⁿ high and low voltage levels.

(*) Types of Computer language:

(i): Machine level language:

A low level language made up of binary numbers or bits that a computer can understand i.e., 0 & 1.

(ii) Assembly level language:

A low level language that uses symbolic equivalent of computer's machine language or, it uses mnemonics to code programs.

(iii) High level language:

The programming language that uses certain syntax to write a program.

Translators must be used to convert high level language to machine language.

(*) Types of Translators

(i) Compiler:

Compiler translates whole program at once and they generate intermediate machine codes.

(ii) Interpreter:

Interpreter translates single line at a time and they never generate intermediate machine codes.

STRUCTURED PROGRAMMING

(*) Problem Solving: Steps:

i) Problem Analysis:

We determine the input and the output desired and check whether the presentation and processing is possible or not.

ii) Algorithm Development & Flow Chart:

- Algorithm: Sequential representation of procedure.
- flow chart: diagrammatic representation of algorithm

iii) Coding:

- Analysis of suitable platform is done and codes are written using high level language.

(iv): Compilation and Execution:

- the written codes are compiled into machine language that occupies certain computer resources as directed and displays the output.
- Executable file has input, does processing and output.


(v): Debugging and testing:

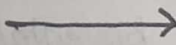
- Errors are checked and the program is tested until the desired output/function is achieved.

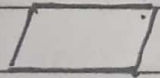
(vi): Program documentation:

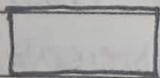
- Information about the program is kept to make other people informed about the program details.


* Structures Used in flow chart:


(i): Oval =  = Terminator = Signalling start/end.

(ii) Arrow =  = gives direction of program flow.

(iii) Parallelogram =  ⇒ showing input and output.

(iv) Rectangle =  ⇒ shows processing step.

(v) Diamond =  = shows decision to be made

(vi) Circle =  ⇒ breakage and connector of flow line.

Q.1: Write algorithm and flow chart for sum of two numbers.

Ans:

Algorithm:

(i): START

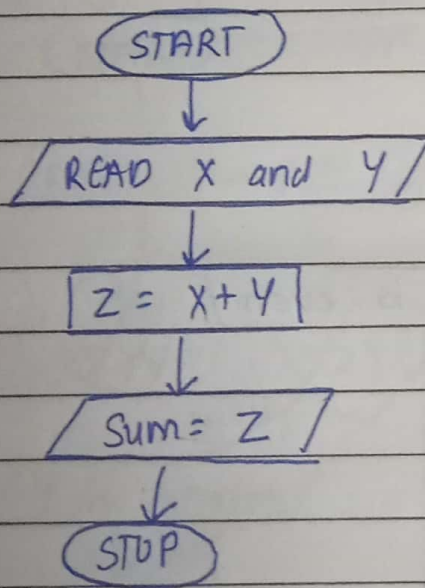
(ii) READ X and Y

(iii) ADD X and Y ie, $X+Y=Z$

(iv) Display sum ie, Z

(v): STOP

Flow chart:



Q. 2: Write algorithm and flow chart to find out the number is odd or even.

Ans:

Algorithm:

i) START

ii) READ X

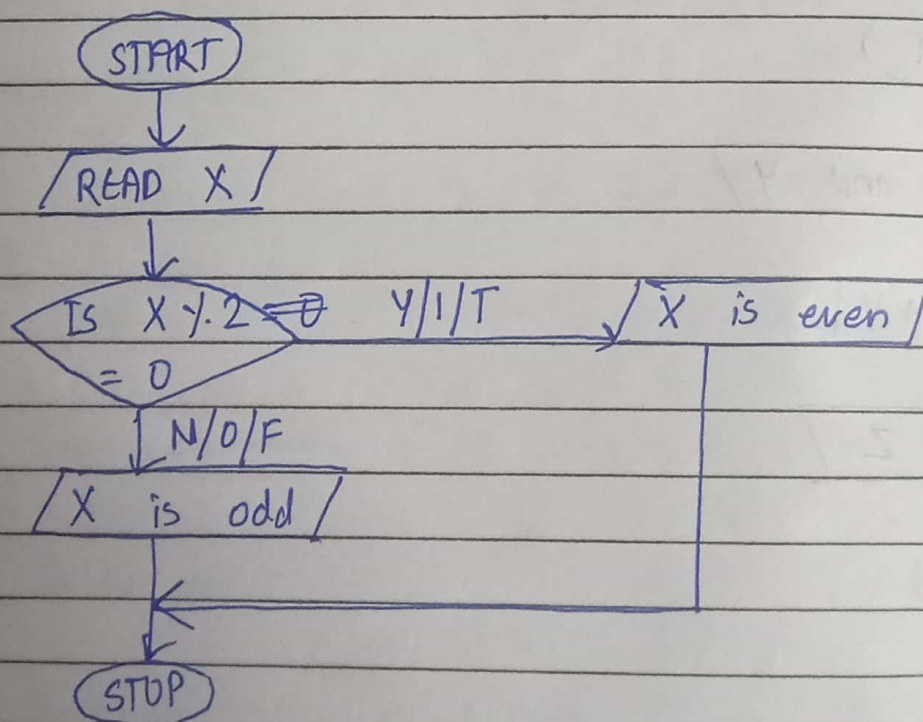
iii) Is X divisible by 2

If yes, output "X is even"

If no, output "X is odd"

iv) STOP

Flow chart:



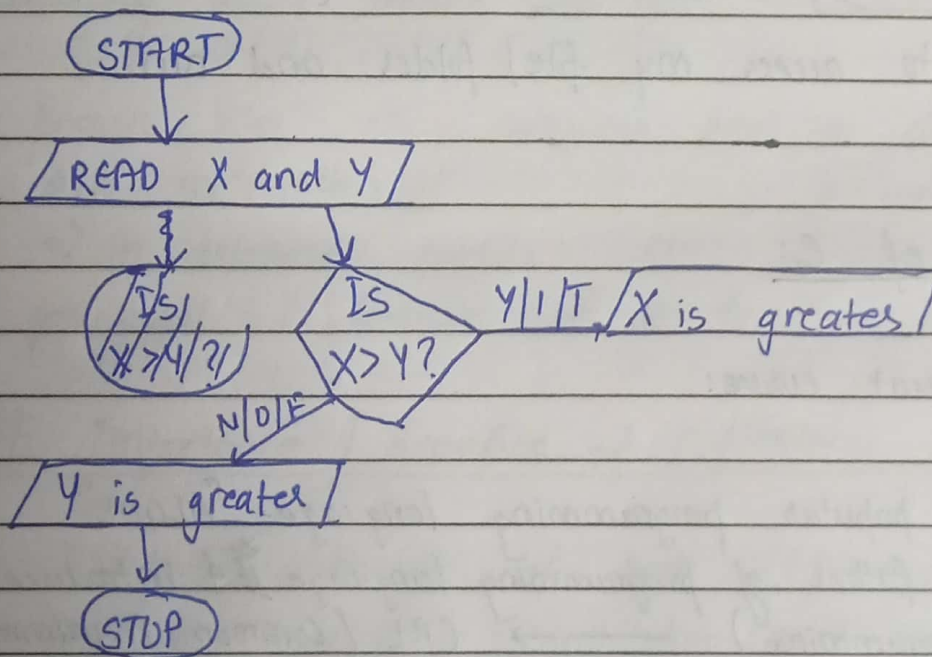
Q.3: Write algorithm and flow chart to find out which number is greater.

Ans:

Algorithm:

- i) START
- ii) READ X and Y
- iii) IS $X > Y$,
 if yes, output "X is greater"
 if no, output "Y is greater"
- iv) STOP

Flow chart:



C - programming:

(*) Terminal functions: (Used in lab):

x) gcc hello.c -o 1
(file name changes to 1)

(i): pwd: present working directory

(ii) ls: listing of files.

(iii) cd ~ ↵: home directory

(iv) cd Desktop ↵: desktop

(v) mkdir: creates folder

(vi) gedit {file name}: ~~translates~~ ^{saves file / opens file}

(vii) gcc = compiler

(viii) ./ (filename) : Run.

(ix) a.out = file created.

Steps to access my file/folder and run

History of C:

(*) Development curve:

1st most popular programming language "ALGOL"
(regarded as father of programming language & introduced structured programming) → CPL (Common Programming language) → BCPL (Basic Combined Programming language)
(created using ALGOL + CPL) → B (used for creating UNIX OS)

Then,

To fully utilize PDP 11 processor and its functions, C programming language was evolved from B programming language by 'Denise Ritchie' in 1970 at Bell Telephone Laboratories.

Introduction to C

- C-programming language is a system programming language (general purpose procedural p.l.)
- C-programming is structured programming language / p.l. (creating hardware-software interface using device drivers.)
- interacts with hardware using hardware-software interface using device drivers.
- presence in OS, compiles and in database.
 - in OS: Unix
 - in database: MySQL, ORACLE
- procedural p.l. because we use functions to do tasks.

Importance / Benefits of C / Features of C

- i) It provides benefits of both LLL and HLL
- ii) It has very less compilation and execution time due to less embedded functions.
- iii) It is develops logic and it is base of all logical ~~problem~~ programming languages.

Desired Program Characteristics:

This also means characteristics of a good program.

- i): **Clarity:** Code must be written clearly and presentation of codes must be properly done.
Eg: don't do all in a single line, giving spaces.
- ii) **Accuracy:** The logic or calculation that is instructed must be kept accurate and must not have double meanings.
- iii) **Simplicity:** As far as possible, the coding must be simple and to the point.
Eg: not including loop in a single program.
- iv) **Efficiency:** For a single program minimal codes should be written such that execution speed is high and effective utilization is possible.
Eg: 10 line code is better than 100 line code for same ~~purpose~~ purpose.
- v) **Modularity:** Programs can be broken into small modules helping to increase clarity and accuracy. ie, module editing is easier than whole program.
- vi) **Generality:** Program written must be general and within reasonable limit. This helps the people other than the developer understand.