# Q1. Design an algorithm and draw a flowchart that will read The two sides of a rectangle and calculate it’s area ?

**And:-**

 Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

Pseudocode

 input the width (W) and Length (L) of a rectangle Calculate the area (A) by multiplying L with W print A

 Algorithm

Step 1: Input W,L

Step 2: A L x W

Step 3: Print A

START

input W, L

A L x W Print A

STOP

Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation

ax²+bx+c=0

B2\_4ac

Hint: d= sqrt ( ), and the roots are: x1= (–b+ d)/2a and x2= (–b–d)/2a

Pseudocode:

putt the coefficients (a, b, c) of the quadratic equation

Calculate d

Calculate x1

Calculate x2

Print x1 and x2

The algorithm for the flowchart is as

 follows.

If A>B then print A

 else print B

 end if

Y is A>B N

Print A BPrint

**Q1(b):-** Name different types of errors which can occur during the execution of a program?

**And:-**

Name different types errors which can occur during the execution of a program.

During the execution of a program, the errors that may occur are:

Syntax errors: errors occur when our program contains grammatical errors

Ex: Suppose we didn't put semicolon at the end of a statement.

Run time errors: These errors occur while the program is running.

Logical errors: Errors such as calculation mistakes.

Runtime Error . Which are also part of errors.

There are three kinds of errors: syntax errors, runtime errors, and logic errors.

These are errors where the compiler finds something wrong with your program, and you can't even try to execute it.

 For example, you may have incorrect punctuation, or may be trying to use a variable that hasn't been declared.

**Q2(a):-**  Design an algorithm that reads two value determine the largest value and prints the largest value with an identified message.

**And:-**

Pseudocode

 input the width (W) and Length (L) of a rectangle Calculate the area (A) by multiplying L with W print A

 Algorithm

Step 1: Input W,L

Step 2: A L x W

Step 3: Print A

START

input W, L

A L x W Print A

STOP

Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation

ax²+bx+c=0

B2\_4ac

Hint: d= sqrt ( ), and the roots are: x1= (–b+ d)/2a and x2= (–b–d)/2a

Pseudocode:

putt the coefficients (a, b, c) of the quadratic equation

Calculate d

Calculate x1

Calculate x2

Print x1 and x2

The algorithm for the flowchart is as

 follows.

If A>B then print A

 else print B

 end if

Y is A>B N

Print A BPrint

**Q2(b):-** What do you understand by the term Maintain and update the program?

**And:-**

Definition:

 Software maintenance is a part of Software Development Life Cycle.

Its main purpose is to modify and update software application after delivery to correct faults and to improve performance.

When the real world changes, the software requires alteration wherever possible.

Software Maintenance Software maintenance is a part of Software Development Life Cycle. Its main purpose is to modify.

Maintain and Update the program .Maintenance and update are the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

An update is new, improved, or fixed software, which replaces older versions of the same software. For example, updating your operating system brings it up-to-date with the latest drivers, system utilities, and security software. Updates are often provided by the software publisher free of additional charge.

* **Q3 :- Differentiate** **between** **the** **following**
* **Ans(A)**

**Bug & Debugg**

As verbs the difference between debug and bug

is that debug is (computer science) to search for and eliminate malfunctioning elements or errors in something, especially a computer program or machinery while bug is (informal transitive) to annoy.

* **Bug**

Noun

(en noun)

An insect of the order Hemiptera (the "true bugs").

(colloquial) Any insect, arachnid, or other terrestrial arthropod that is a pest.

 These flies are a bother. I’ll get some bug spray and kill them.

Various species of marine or freshwater crustaceans; e.g. a Morton Bay bug, mudbug.

A problem that needs fixing, especially in computing.

The software bug led the computer to calculate 2 plus 2 as 5.

* **Debug:-**

Verb (debugg)

(computer science) To search for and eliminate malfunctioning elements or errors in something, especially a computer program or machinery.

(electronics) To remove a hidden electronic surveillance device from (somewhere).

(US) To remove insects from (somewhere).

diagnose troubleshoot.

* **Ans3(B):-**
* **Syntex error** & **logical** **error:-**

There are generally two types of errors: syntax errors and logic errors. Syntax errors occur when a program does not conform to the grammar of a programming language, and the compiler cannot compile the source file.

* Syntex Error:-

Syntax errors occur when a program does not conform to the grammar of a programming language, and the compiler cannot compile the source file. Logic errors occur when a program does not do what the programmer expects it to do. Syntex error occur when program does not confirmer.

* Logical Error:-

Logic error compiled when language are interpreted Logic errors occur in both compiled and interpreted languages. Unlike a program with a syntax error, a program with a logic error is a valid program in the language, though it does not behave as intended. Often the only clue to the existence of logic errors is the production of wrong solutions, though static analysis may sometimes spot them.

logic error is a bug in a program that causes it to operate incorrectly, but not to terminate abnormally (or crash). A logic error produces unintended or undesired output or other behaviour, although it may not immediately be recognized as such.

* **Ans3(c)**. **Compiler** & **Assembler:-**

**Compiler :-** A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. Typically, a programmer writes language statements in a language such as Pascal or C one line at a time using an editor. The programing language turn into machine language.

Compiler converts the source code written into by the programmer to a machine level language. Assembler converts the assembly code into the machine code. It converts the whole code into machine language at a time. But the Assembler can't do this at once.

**Assembler:-**

All assembler are simple compiler since they transfer from one language to another language. An assembler is a compiler which performs a specific set of tasks. The terms have somewhat diverged in practice, but the basic definition of "compiler" (translate between languages)

All assemblers are (simple) compilers, since they transform one language to another. Not all of compilers are assemblers.

It converts a source code to an object code first and then it converts the object code to the machine language with the help of the

Input to the assembler is assembly language code.

At the output of assembler is re-locatable machine code generated by an assembler is represented by binary code.

It produces either an assembly language programme, as an intermediate step, or else machine code directly. Either way, the final result is machine code.It is difficult to debugging.

**And3(d):- System software & Application software:-**

**System Software:-**

System software is software designed to provide a platform for other software.

 Examples of system software include operating systems like macOS, GNU/Linux and Microsoft Windows, computational science software, game engines, industrial automation, and software as a service applications. System software is an a software of an a system.

System software refers to the files and programs that make up your computer's operating system, and also System files include.

System software is a type of computer program that is designed to run a computer's hardware and application programs. If we think of the computer system as a layered model, the system software is the interface between the hardware and user applications. These are all about software of a system

**Application Software:-**

Application software is a type of computer program that performs a specific personal, educational, and business function. These type of application Software are mostly use for business , education, and personally.

Application software is a term which is used for software created for a specific purpose. It is generally a program or collection of programs used by end users.

Application software, or app for short, is software that performs specific tasks for an end-user.

System software is a type of computer program that is designed to run a computer's hardware and application programs.

If we think of the computer system as a layered model, the system software is the interface between the hardware and user applications.

**Ans3(e):- low level language & high level language:-**

Low level language is machine readable form of program. Whereas the high level language will be in human readable form.

 High level language uses compilers and interpreters which requires large memory space. High level language require large memory of spaces.

**High-level** **Languages:-**

High-Level Languages are easy to learn and understand.

They are executed slower than lower level languages because they require a translator program.

They allow much more abstractions.

They do not provide many facilities at the hardware levels.

For writing programs, hardware knowledge is not required.

The programs are easy to modified

A single statement may execute several instructions.

BASIC, Perl, Pascal, COBOL, Ruby etc are examples of High-Level Languages.

**Low\_level language:-**

Low-Level Languages

Low-Level Languages are challenging to learn and understand.

They execute with high speed.

They allow little or no abstraction.

They are very close to the hardware and help to write a program at the hardware level.

For writing a programs, hardware knowledge is a must.

Modifying programs is difficult.

The statements are can be directly mapped to processor instructions.

Machine language and Assembly language are Low-Level Languages.