SURA'S

COMPUTER SCIENCE

12th Standard

Public Exam Edition 2021-22

Strictly as per the Reduced (Prioritised) Syllabus released on 13th August, 2021 (G.O.(Ms).No126)

Salient Features

- Complete Solutions to Textbook Exercises.
- Model Question Papers 1 to 6 (PTA) : Questions are incorporated in the appropriate sections.
- Govt. Model Question Paper - 2019 (Govt. MQP-2019), Quarterly Exam - 2019 (QY-2019), Half Yearly Exam - 2019 (HY-2019), Public Exam March - 2020 (Mar-2020) and Govt. Supplementary Exam - Sep - 2020 (Sep-2020) are incorporated in the appropriate sections.
- Sura's Model question paper with answers are given based on the reduced syllabus.

SURA PUBLICATIONS
Chennai

2021-22 Edition

ISBN : 978-93-5330-423-2

Code No : RPS_020

Author :

Mr. Shanmugasundaram

(Post Graduate Teacher, Chennai)

Edited by :

Mrs. Malathy Krishnamoorthy M.Sc.,

Erode

Reviewed by :

Mr. Balaji M.Sc., M.Phil.

Chennai

Head Office:

1620, 'J' Block, 16th Main Road,

Anna Nagar, **Chennai - 600 040.**

Phone: 044-4862 9977, 044-486 27755.

Mob : 81242 01000/ 81243 01000

Fax : (91) 44-26162173

e-mail : orders @surabooks.com

website : www.surabooks.com

Strictly as per the Reduced (Prioritised) Syllabus released on 13th August, 2021 (G.O.(Ms).No126)

PRACTICALS	
CLASS: 12	SUBJECT: COMPUTER SCIENCE
Sl.No	Topic
1	PY1(a) Calculate Factorial PY1(b) Sum of Series
2	PY2(a) Odd or Even PY2(b)Reverse the String
3	PY3 Generate values and remove odd numbers
4	PY4 Generate Prime numbers and Set Operations
5	PY5 Display a String elements – Using Class

PREFACE

" The woods are lovely, dark and deep. "
But I have promises to keep, and
miles to go before I sleep

- Robert Frost

Respected Principals, Correspondents, Head Masters / Head Mistresses, Teachers,

From the bottom of our heart, we at SURA Publications sincerely thank you for the support and patronage that you have extended to us for more than a decade.

It is in our sincerest effort we take the pride of releasing **SURA's Computer Science Guide** for +2 Standard – Edition 2021-22. This guide has been authored and edited by qualified teachers having teaching experience for over a decade in their respective subject fields. This Guide has been reviewed by reputed Professors who are currently serving as Head of the Department in esteemed Universities and Colleges.

With due respect to Teachers, I would like to mention that this guide will serve as a teaching companion to qualified teachers. Also, this guide will be an excellent learning companion to students with exhaustive exercises and in-text questions in addition to precise answers for textual questions.

In complete cognizance of the dedicated role of Teachers, I completely believe that our students will learn the subject effectively with this guide and prove their excellence in Board Examinations.

I once again sincerely thank the Teachers, Parents and Students for supporting and valuing our efforts.

God Bless all.

Subash Raj, B.E., M.S.
- Publisher
Sura Publications

All the Best

CONTENTS

TO ORDER WITH US

SCHOOLS and TEACHERS

We are grateful for your support and patronage to **'SURA PUBLICATIONS'**
Kindly prepare your order in your School letterhead and send it to us.
For Orders contact: 81242 01000 / 81243 01000

DIRECT DEPOSIT

A/c Name	: **Sura Publications**
Our A/c No.	: **36550290536**
Bank Name	: **STATE BANK OF INDIA**
Bank Branch	: PADI
IFSC	: SBIN0005083

A/c Name	: **Sura Publications**
Our A/c No.	: **21000210001240**
Bank Name	: **UCO BANK**
Bank Branch	: Anna Nagar West
IFSC	: UCBA0002100

A/c Name	: **Sura Publications**
Our A/c No.	: **6502699356**
Bank Name	: **INDIAN BANK**
Bank Branch	: ASIAD COLONY
IFSC	: IDIB000A098

A/c Name	: **Sura Publications**
Our A/c No.	: **1154135000017684**
Bank Name	: **KVB BANK**
Bank Branch	: Anna Nagar
IFSC	: KVBL0001154

After Deposit, please send challan and order to our address.
email : orders@surabooks.com / Whatsapp : 81242 01000.

DEMAND DRAFT / CHEQUE

Please send Demand Draft / cheque in favour of **'SURA PUBLICATIONS'** payable at **Chennai**.

The Demand Draft / cheque should be sent with your order in School letterhead.

STUDENTS

Order via Money Order (M/O) to

SURA PUBLICATIONS
1620, 'J' Block, 16th Main Road, Anna Nagar,
Chennai - 600 040.
Phone : 044 - 4862 9977, 044 - 486 27755.
Mobile : 96001 75757/ 81242 01000/81243 01000.
email : orders@surabooks.com Website : www.surabooks.com

UNIT-I PROBLEM SOLVING TECHNIQUES

CHAPTER 1

FUNCTION

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

1. The small sections of code that are used to perform a particular task is called

(a) Subroutines (b) Files
(c) Pseudo code (d) Modules

[Ans. (a) Subroutines]

2. Which of the following is a unit of code that is often defined within a greater code structure?

(a) Subroutines (b) Function
(c) Files (d) Modules

[Ans. (b) Function]

3. Which of the following is a distinct syntactic block? *[PTA-6]*

(a) Subroutines (b) Function
(c) Definition (d) Modules

[Ans. (c) Definition]

4. The variables in a function definition are called as *[PTA-2; QY-2019]*

(a) Subroutines (b) Function
(c) Definition (d) Parameters

[Ans. (d) Parameters]

5. The values which are passed to a function definition are called *[HY-2019]*

(a) Arguments (b) Subroutines
(c) Function (d) Definition

[Ans. (a) Arguments]

6. Which of the following are mandatory to write the type annotations in the function definition? *[PTA-4]*

(a) Curly braces (b) Parentheses
(c) Square brackets (d) Indentations

[Ans. (b) Parentheses]

PART - II

ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

1. What is a subroutine? *[PTA-1; HY-2019]*

Ans. (i) Subroutines are the basic building blocks of computer programs. Subroutines are small sections of code that are used to perform a particular task that can be used repeatedly.

(ii) In Programming languages these subroutines are called as Functions.

2. Define Function with respect to Programming language.

Ans. A function is a unit of code that is often defined within a greater code structure. Specifically, a function contains a set of code that works on many kinds of inputs, like variants, expressions and produces a concrete output.

5. Which of the following is a normal function definition and which is recursive function definition.

i)
```
let rec sum x y :
return x + y
```

ii)
```
let disp :
print 'welcome'
```

iii)
```
let rec sum num :
if (num!=0) then return num + sum (num-1)
else
return num
```

Ans. (i) Recursive function
(ii) Normal function
(iii) Recursive function

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. What are called Parameters and write a note on *[PTA-2]*

(i) Parameter without Type

(ii) Parameter with Type

Ans. **Parameters (and arguments) :** Parameters are the variables in a function definition and arguments are the values which are passed to a function definition.

(i) **Parameter without Type :** Let us see an example of a function, definition :

```
(requires: b>=0 )
(returns: a to the power of b)
let rec pow a b:=
        if b=0 then 1
        else a * pow a (b –1)
```

- In the above function definition variable 'b' is the parameter and the value which is passed to the variable 'b' is the argument. The precondition **(requires)** and postcondition **(returns)** of the function is given.
- Note we have not mentioned any types: **(data types)**. Some language compiler solves this type **(data type)** inference problem algorithmically, but some require the type to be mentioned.
- In the above function definition if expression can return 1 in the then branch, by the **typing** rule the entire if expression has type **int**.
- Since the if expression has type **'int'**, the function's return type also be 'int'. **'b'** is compared to 0 with the equality operator, so **'b'** is also a type of 'int'. Since 'a' is multiplied with another expression using the * operator, **'a'** must be an int.

(ii) **Parameter with Type :** Now let us write the same function definition with types for some reason:

```
(requires: b> 0 )
(returns: a to the power of b )
    let rec pow (a: int) (b: int) : int :=
            if b=0 then 1
            else a * pow b (a-1)
```

- When we write the type annotations for **'a'** and **'b'** the parentheses are mandatory. Generally we can leave out these annotations, because it's simpler to let the compiler infer them.
- There are times we may want to explicitly write down types. This is useful on times when you get a type error from the compiler that doesn't make sense. Explicitly annotating the types can help with debugging such an error message.

2. Identify in the following program *[PTA-5]*

```
let rec gcd a b :=
if b <> 0 then gcd b (a mod b) else return a
```

i) Name of the function

ii) Identify the statement which tells it is a recursive function

iii) Name of the argument variable

iv) Statement which invoke the function recursively

v) Statement which terminates the recursion

Ans. (i) gcd
(ii) let rec gcd
(iii) a, b
(iv) gcd b (a mod b)
(v) return a

HANDS ON PRACTICE

1. Write algorithmic function definition to find the minimum among 3 numbers.

Ans.

```
let min 3 x y z :=
        if x < y then
                if x < z then x else z
        else
                if y < z then y else z
```

2. Write algorithmic recursive function definition to find the sum of n natural numbers.

Ans.

```
let rec sum num:
        if (num!=0) then return num+sum num-1)
        else
        return num
```

PTA Questions and Answers

1 MARK

1. A function definition which call itself : *[PTA-1]*

(a) Pure function (b) Impure function
(c) Normal function
(d) Recursive function

[Ans. (d) Recursive function]

3 MARKS

1. Write a function that finds the minimum of its three arguments. *[PTA-4; QY-2019]*

Ans.
```
let min 3 x y z :=
        if x < y then
            if x < z then x else z
        else
            if y < z then y else z
```

Government Exam Questions and Answers

1 MARK

1. Which is the basic building block of computer programs? *[Sep-2020]*

(a) Argument (b) Parameter
(c) Subroutine (d) Interface

[Ans. (c) Subroutine]

CHAPTER 2 DATA ABSTRACTION

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

1. **Which of the following functions that build the abstract data type ?** *[Sep-2020]*
   (a) Constructors (b) Destructors
   (c) Recursive (d) Nested
   [Ans. (a) Constructors]

2. **Which of the following functions that retrieve information from the data type?**
   (a) Constructors (b) Selectors
   (c) Recursive (d) Nested
   [Ans. (b) Selectors]

6. **The data type whose representation is unknown are called**
   (a) Built in datatype (b) Derived datatype
   (c) Concrete datatype (d) Abstract datatype
   [Ans. (d) Abstract datatype]

PART - II

ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

1. **What is abstract data type?**

Ans. (i) Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of value and a set of operations.

(ii) The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.

2. **Differentiate constructors and selectors.**

Ans. *[PTA-2, 3; QY-2019]*

S. No.	Constructors	Selectors
(i)	Constructors are functions that build the abstract data type.	Selectors are functions that retrieve information from the data type.
(ii)	Constructors create an object, bundling together different pieces of information.	Selectors extract individual pieces of information from the object

PART - III

ANSWER THE FOLLOWING QUESTIONS (3 MARKS)

3. **Identify Which of the following are constructors and selectors?** *[PTA-5]*
   (a) N1=number()
   (b) accetnum(n1)
   (c) displaynum(n1)
   (d) eval(a/b)
   (e) x,y= makeslope (m), makeslope(n)
   (f) display()

Ans. (a) Constructors
(b) Selectors
(c) Selectors
(d) Selectors
(e) Constructors
(f) Selectors

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. How will you facilitate data abstraction. Explain it with suitable example. *[PTA-2, 4]*

Ans. Data abstraction is supported by defining an abstract data type (ADT), which is a collection of constructors and selectors. To facilitate data abstraction, you will need to create two types of functions: **Constructors, Selectors**

Constructors :

(i) Constructors are functions that build the abstract data type.

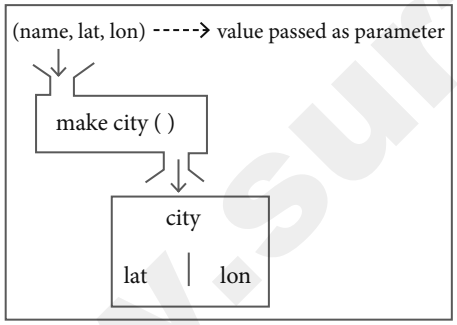
(ii) Constructors create an object, bundling together different pieces of information.

(iii) For example, say you have an abstract data type called city.

(iv) This city object will hold the city's name, and its latitude and longitude.

(v) To create a city object, you'd use a function like **city = makecity (name, lat, lon)**.

(vi) Here makecity (name, lat, lon) is the constructor which creates the object city.



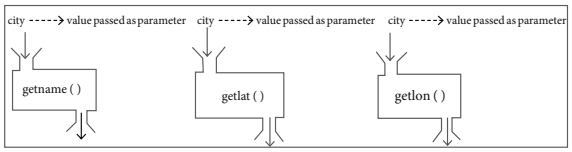
Constructor

Selectors :

(i) Selectors are functions that retrieve information from the data type.

(ii) Selectors extract individual pieces of information from the object.

(iii) To extract the information of a city object, you would used functions like
getname(city)
getlat(city)
getlon(city)
These are the selectors because these functions extract the information of the city object.



PTA QUESTIONS AND ANSWERS

1 MARK

1. Expansion of ADT : *[PTA-1]*

(a) Abstract Data Tuple
(b) All Data Template
(c) Abstract Data Type
(d) Application Data Type

[Ans. (c) Abstract Data Type]

2. ADT can be implemented using ____. *[PTA-5]*

(a) singly linked list (b) doubly linked list
(c) either A or B (d) neither A nor B

[Ans. (a) singly linked list]

GOVERNMENT EXAM QUESTIONS AND ANSWERS

1 MARK

1. The datatype whose representation is unknown is called *[HY-2019]*

(a) Built-in datatype (b) Derived datatype
(c) Concrete datatype (d) Abstract datatype

[Ans. (d) Abstract datatype]

3 MARKS

1. (a) What is selector? *[Sep-2020]*

(b) What are the parts of a program?

Ans. a) Selectors are nothing but the functions that retrieve information from the data type. Therefore in the above code

(i) getname(city)

(ii) getlat(city)

(iii) getlon(city)
are the selectors because these functions extract the information of the city object

b) The two parts of a program are, the part that operates on abstract data and the part that defines a concrete representation, is connected by a small set of functions that implement abstract data in terms of the concrete representation.

CHAPTER 3

SCOPING

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

1. **Which of the following refers to the visibility of variables in one part of a program to another part of the same program.**
   (a) Scope (b) Memory
   (c) Address (d) Accessibility
   [Ans. (a) Scope]

2. **The process of binding a variable name with an object is called** *[Sep-2020]*
   (a) Scope (b) Mapping
   (c) late binding (d) early binding
   [Ans. (b) Mapping]

3. **Which of the following is used in programming languages to map the variable and object?** *[PTA-2; HY-2019]*
   (a) :: (b) :=
   (c) = (d) ==
   [Ans. (c) =]

4. **Containers for mapping names of variables to objects is called** *[QY-2019]*
   (a) Scope (b) Mapping
   (c) Binding (d) Namespaces
   [Ans. (d) Namespaces]

5. **Which scope refers to variables defined in current function?**
   (a) Local Scope (b) Global scope
   (c) Module scope (d) Function Scope
   [Ans. (a) Local Scope]

PART - II

ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

1. **What is a scope?**

Ans. Scope refers to the visibility of variables, parameters and functions in one part of a program to another part of the same program.

2. **Why scope should be used for variable. State the reason.**

Ans. Essentially, variables are addresses (references, or pointers), to an object in memory. When you assign a variable with := to an instance (object), you're binding (or mapping) the variable to that instance. Multiple variable can be mapped to the same instance.

3. **What is Mapping?** *[PTA-5]*

Ans. The process of binding a variable name with an object is called mapping.= (equal to sign) is used in programming languages to map the variable and object.

4. **What do you mean by Namespaces?** *[Govt. MQP-2019; PTA-4; Mar.-2020]*

Ans. Namespaces are containers for mapping names of variables to objects.

Example : a : = 5

Here the variable 'a' is mapped to the value '5'.

PART - III

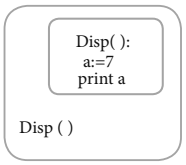
ANSWER THE FOLLOWING QUESTIONS (3 MARKS)

1. **Define Local scope with an example.**

Ans. (i) Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope.

(ii) Only if it does not find it there, the outer scopes are checked.

(iii) Look at this example :

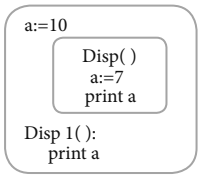
1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program 	Output of the Program 7

(iv) On execution of the above code the variable a displays the value 7, because it is defined and available in the local scope.

2. Define Global scope with an example. *[PTA-6]*

Ans. (i) A variable which is declared outside of all the functions in a program is known as Global variable.

(ii) This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

1. a:=10 2. Disp(): 3. a:=7 4. print a 5. Disp() 6. print a	Entire program 	Output of the Program 7 10

(iii) On execution of the above code the variable a which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because a is defined in global scope.

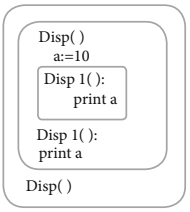
3. Define Enclosed scope with an example.

[PTA-3]

Ans. (i) All programming languages permit functions to be nested. A function (method) within another function is called nested function.

(ii) A variable which is declared inside a function which contains another function definition with in it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.

(iii) When a compiler or interpreter search for a variable in a program, it first search Local, and then search Enclosing scopes. Consider the following example

1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	Entire program 	Output of the Program 10 10

4. Why access control is required?

[PTA-1; HY-2019]

Ans. (i) Access control is a security technique that regulates who or what can view or use resources in a computing environment.

(ii) It is a fundamental concept in security that minimizes risk to the object.

(iii) In other words access control is a selective restriction of access to data.

(iv) In oops Access control is implemented through access modifiers.

5. Identify the scope of the variables in the following pseudo code and write its output

```
color:= Red
mycolor():
b:=Blue
lue
myfavcolor():
    g:=Green
    printcolor, b, g
    myfavcolor()
    printcolor, b
mycolor()
print color
```

Ans. **Output :**

Red Blue Green
Red Blue
Red

Scope of Variables :

Variables	Scope
Color:=Red	Global
b:=Blue	Enclosed
G:=Green	Local

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. Explain the types of scopes for variable or LEGB rule with example. *[PTA-1; Sep-2020]*

Ans. **Types of Variable Scope :**

There are 4 types of Variable Scope, let's discuss them one by one:

Local Scope :

(i) Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope. Only if it does not find it there, the outer scopes are checked.

Look at this example

1. Disp(): 2. a:=7 3. print a 4. Disp()	Entire program Disp( ): a:=7 print a Disp ( )	Output of the Program 7

(ii) On execution of the above code the variable a displays the value 7, because it is defined and available in the local scope.

Global Scope:

(i) A variable which is declared outside of all the functions in a program is known as global variable.

(ii) This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

1. a:=10 2. Disp(): 3. a:=7 4. print a 5. Disp() 6. print a	Entire program a:=10 Disp( ) a:=7 print a Disp 1( ): print a	Output of the Program 7 10

(iii) On execution of the above code the variable 'a' which is defined inside the function displays the value 7 for the function call Disp() and then it displays 10, because a is defined in global scope.

Enclosed Scope :

(i) All programming languages permit functions to be nested. A function (method) with in another function is called nested function.

(ii) A variable which is declared inside a function which contains another function definition with in it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.

(iii) When a compiler or interpreter search for a variable in a program, it first search Local, and then search Enclosing scopes. Consider the following example

1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	Entire program Disp( ) a:=10 Disp 1( ): print a Disp 1( ): print a Disp( )	Output of the Program 10 10

(iv) In the above example Disp1() is defined with in Disp(). The variable 'a' defined in Disp() can be even used by Disp1() because it is also a member of Disp().

Built-in Scope :

(i) The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter.

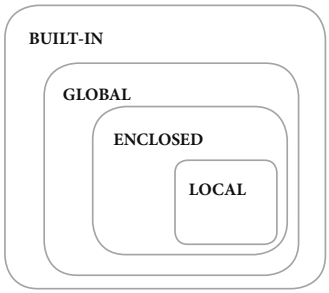
(ii) Any variable or module which is defined in the library functions of a programming language has Built-in or module scope. Consider the following example.

Entire program Built in/module scope Disp( ) Disp 1( ): print a Disp 1( ): print a Disp( )	Library files associated with the software

LEGB rule :

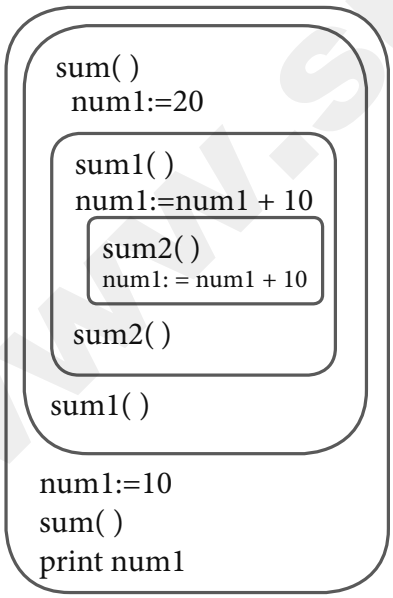
The **LEGB** rule is used to decide the order in which the scopes are to be searched for scope resolution. The scopes are listed below in terms of hierarchy (highest to lowest).

Local(L)	Defined inside function/ class
Enclosed(E)	Defined inside enclosing functions (Nested function concept)
Global(G)	Defined at the uppermost level
Built-in(B)	Reserved names in built-in functions (modules)



Hands on Practice

1. Observe the following diagram and Write the pseudo code for the following.



Ans. sum():

```
num 1:=20
  sum1()
      num1:= num1 + 10
sum2()
      num1:= num1 + 10
  sum2()
sum1()
num1:= 10
sum()
Print num 1
```

PTA Questions and Answers

1 MARK

1. A variable which is declared inside a function which contains another function definition : *[PTA-1]*

(a) Local (b) Global
(c) Enclosed (d) Built-in

[Ans. (c) Enclosed]

2. Which are loaded as soon as the library files are imported to the program? *[PTA-3]*

(a) Built-in scope variables
(b) Enclosed scope variables
(c) Global scope variables
(d) Local scope variables

[Ans. (a) Built-in scope variables]

Government Exam Questions and Answers

1 MARK

1. The kind of scope of the variable 'a' used in the pseudo code given below. *[Govt. MQP-2019]*

(a) Disp(): (b) a: = 7
(c) print a (d) Disp()
(a) Local (b) Global
(c) Enclosed (d) Built-in

[Ans. (a) Local]

2 MARKS

1. What is LEGB rule? *[QY-2019]*

Ans. Scope also defines the order in which variables have to be mapped to the object in order to obtain the value.

CHAPTER 4 ALGORITHMIC STRATEGIES

EVALUATION

PART - I

CHOOSE THE BEST ANSWER (1 MARK)

1. The word comes from the name of a Persian mathematician Abu Ja'far Mohammed ibn-i Musa al Khowarizmi is called? *[PTA-6]*

(a) Flowchart (b) Flow
(c) Algorithm (d) Syntax

[Ans. (c) Algorithm]

2. From the following sorting algorithms which algorithm needs the minimum number of swaps?

(a) Bubble sort (b) Quick sort
(c) Merge sort (d) Selection sort

[Ans. (d) Selection sort]

4. The complexity of linear search algorithm is

(a) O(n) (b) O(log n)
(c) O(n2) (d) O(n log n)

[Ans. (a) O(n)]

5. From the following sorting algorithms which has the lowest worst case complexity?

(a) Bubble sort (b) Quick sort
(c) Merge sort (d) Selection sort

[Ans. (c) Merge sort]

6. Which of the following is not a stable sorting algorithm?

(a) Insertion sort (b) Selection sort
(c) Bubble sort (d) Merge sort

[Ans. (b) Selection sort]

7. Time complexity of bubble sort in best case is *[PTA-1]*

(a) θ (n) (b) θ (nlogn)
(c) θ (n2) (d) θ (n(logn) 2)

[Ans. (a) θ (n)]

9. If a problem can be broken into subproblems which are reused several times, the problem possesses which property?

(a) Overlapping subproblems
(b) Optimal substructure
(c) Memoization
(d) Greedy

[Ans. (a) Overlapping subporblems]

PART - II

ANSWER THE FOLLOWING QUESTIONS (2 MARKS)

1. What is an Algorithm? *[Mar.-2020]*

Ans. An algorithm is a finite set of instructions to accomplish a particular task. It is a step-by-step procedure for solving a given problem.

2. Define Pseudo code.

Ans. (i) Pseudo code is an informal high level description of the operations principle of a computer program or other algorithm.

(ii) It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

3. Who is an Algorist?

Ans. Algorist may refer to

(i) A person skilled in the technique of performing basic decimal arithmetic, known as algorism.

(ii) A person skilled in the design of algorithms.

(iii) An Algorithmic artist.

4. What is Sorting?

Ans. Sorting is any process of arranging information or data in an ordered sequence either in

ascending or descending order. Various sorting techniques in algorithms are Bubble sort, Quick sort, Heap sort, Selection sort, Insertion sort.

5. What is searching? Write its types.

[Govt. MQP-2019; HY-2019]

Ans. A searching algorithm is the step-by-step procedure used to locate specific data among a collection of data. There are two type of searching are

(i) Linear Search

(ii) Binary Search

PART - III

ANSWER THE FOLLOWING QUESTIONS

(3 MARKS)

1. List the characteristics of an algorithm.

Ans. (i) Input

(ii) Output

(iii) Finiteness

(iv) Definiteness

(v) Effectiveness

(vi) Correctness

(vii) Simplicity

(viii) Unambiguous

(ix) Feasibility

(x) Portable

(xi) Independent

PART - IV

ANSWER THE FOLLOWING QUESTIONS

(5 MARKS)

1. Explain the characteristics of an algorithm.

Ans. *[PTA-5; HY-2019]*

Input	Zero or more quantities to be supplied.
Output	At least one quantity is produced.
Finiteness	Algorithms must terminate after finite number of steps.
Definiteness	All operations should be well defined. For example operations involving division by zero or taking square root for negative number are unacceptable.
Effectiveness	Every instruction must be carried out effectively.
Correctness	The algorithms should be error free.
Simplicity	East to implement.
Unambiguous	Algorithm should be clear and unambiguous. Each of its steps and their inputs/outputs should be clear and must lead to only one meaning.
Feasibility	Should be feasible with the available resources.
Portable	An algorithm should be generic, independent of any programming language or an operating system able to handle all range of inputs.
Independent	An algorithm should have step-by-step directions, which should be independent of any programming code.

2. Discuss about Linear search algorithm.

[PTA-1; Mar.-2020]

Ans. (i) Linear search also called sequential search is a sequential method for finding a particular value in a list.

(ii) This method checks the search element with each element in sequence until the desired element is found or the list is exhausted. In this searching algorithm, list need not be ordered.

Pseudo code :

(i) Traverse the array using for loop

(ii) In every iteration, compare the target search key value with the current value of the list.

- If the values match, display the current index and value of the array
- If the values do not match, move on to the next array element.

(iii) If no match is found, display the search element not found.

Example

To search the number 25 in the array given below, linear search will go step by step in a sequential order starting from the first element in the given array if the search element is found that index is returned otherwise the search is

continued till the last index of the array. In this example number 25 is found at index number 3.

index	0	1	2	3	4
values	10	12	20	25	30

Example 1 :

Input: values[] = {5, 34, 65, 12, 77, 35}

target = 77

Output: 4

Example 2:

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}

target = 200

Output: -1 (not found)

3. What is Binary search? Discuss with example.

Ans. **Binary search :** Binary search also called half-interval search algorithm. It finds the position of a search element within a sorted array. The binary search algorithm can be done as divide-and-conquer search algorithm and executes in logarithmic time.

Pseudo code for Binary search :

Start with the middle element:

(i) If the search element is equal to the middle element of the array i.e., the middle value = number of elements in array/2, then return the index of the middle element.

(ii) If not, then compare the middle element with the search value,

(iii) If the search element is greater than the number in the middle index, then select the elements to the right side of the middle index, and go to Step-1.

(iv) If the search element is less than the number in the middle index, then select the elements to the left side of the middle index, and start with Step-1.

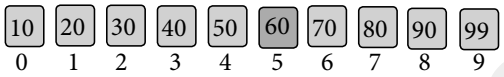
(v) When a match is found, display success message with the index of the element matched.

(vi) If no match is found for all comparisons, then display unsuccessful message.

Binary Search Working principles :

(i) List of elements in an array must be sorted first for Binary search. The following example describes the step by step operation of binary search.

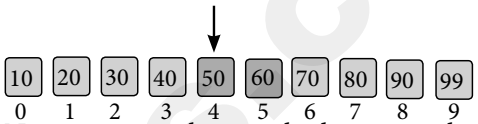
(ii) Consider the following array of elements, the array is being sorted so it enables to do the binary search algorithm. Let us assume that the search element is 60 and we need to search the location or index of search element 60 using binary search.



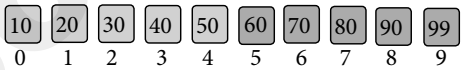
(iii) First, we find index of middle element of the array by using this formula :

mid = low + (high - low) / 2

(iv) Here it is, 0 + (9 - 0) / 2 = 4 (fractional part ignored). So, 4 is the mid value of the array.



(v) Now compare the search element with the value stored at mid value location 4. The value stored at location or index 4 is 50, which is not match with search element. As the search value 60 is greater than 50.

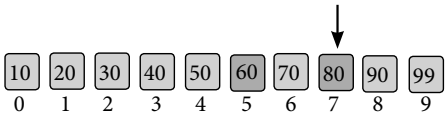


(vi) Now we change our low to mid + 1 and find the new mid value again using the formula.

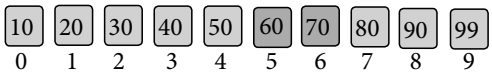
low to mid + 1

mid = low + (high - low) / 2

(vii) Our new mid is 7 now. We compare the value stored at location 7 with our target value 31.



(viii) The value stored at location or index 7 is not a match with search element, rather it is more than what we are looking for. So, the search element must be in the lower part from the current mid value location

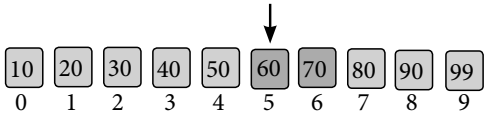


(ix) The search element still not found. Hence, we calculated the mid again by using the formula.

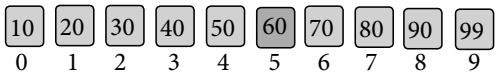
high = mid -1

mid = low + (high - low)/2

Now the mid value is 5.



(x) Now we compare the value stored at location 5 with our search element. We found that it is a match.



(xi) We can conclude that the search element 60 is found at location or index 5. For example if we take the search element as 95, For this value this binary search algorithm return unsuccessful result.

4. Explain the Bubble sort algorithm with example. *[PTA-6]*

Ans. **Bubble sort algorithm:**

(i) Bubble sort is a simple sorting algorithm. The algorithm starts at the beginning of the list of values stored in an array. It compares each pair of adjacent elements and swaps them if they are in the unsorted order.

(ii) This comparison and passed to be continued until no swaps are needed, which indicates that the list of values stored in an array is sorted. The algorithm is a comparison sort, is named for the way smaller elements "bubble" to the top of the list.

(iii) Although the algorithm is simple, it is too slow and less efficient when compared to insertion sort and other sorting methods.

(iv) Assume list is an array of n elements. The swap function swaps the values of the given array elements.

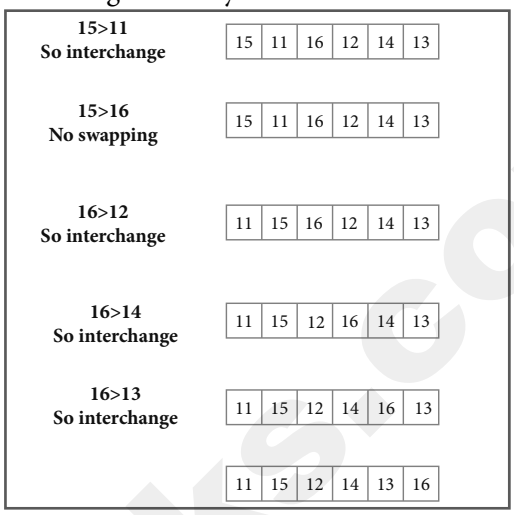
Pseudo code :

(i) Start with the first element i.e., index = 0, compare the current element with the next element of the array.

(ii) If the current element is greater than the next element of the array, swap them.

(iii) If the current element is less than the next or right side of the element, move to the next element. Go to Step 1 and repeat until end of the index is reached.

(iv) Let's consider an array with values {15, 11, 16, 12, 14, 13} Below, we have a pictorial representation of how bubble sort will sort the given array.



(v) The above pictorial example is for iteration–1. Similarly, remaining iteration can be done. The final iteration will give the sorted array. At the end of all the iterations we will get the sorted values in an array as given below :

11	12	13	14	15	16

PTA Questions and Answers

1 MARK

1. Step by step procedure for solving a given problem: *[PTA-2]*

(a) Proram (b) Pseudo Code
(c) Flowchart (d) Algorithm

[Ans. (d) Algorithm]

2. Which of the following is not a characteristic of an algorithm? *[PTA-3]*

(a) Input (b) Program
(c) Finiteness (d) Simplicity

[Ans. (b) Program]

3. This is a theoretical performance analysis of an algorithm. *[PTA-4]*

(a) Priori estimates (b) Posteriori testing
(c) Space factor (d) Time factor

[Ans. (a) Priori estimates]

3 MARKS

1. Write a pseudo code for bubble sort algorithm. *[PTA-3]*

Ans. (i) Start with the first element i.e., index = 0, compare the current element with the next element of the array.

(ii) If the current element is greater than the next element of the array, swap them.

(iii) If the current element is less than the next or right side of the element, move to the next element. Go to Step 1 and repeat until end of the index is reached.

2. Write the pseudo code for linear search. *[PTA-4]*

Ans. (i) Traverse the array using 'for loop'

(ii) In every iteration, compare the target search key value with the current value of the list.

(iii) If the values match, display the current index and value of the array

(iv) If the values do not match, move on to the next array element.

(v) If no match is found, display the search element not found.

3. What are the different phases of analysis and performance evaluation of an algorithm? *[PTA-5]*

Ans. Analysis of algorithms and performance evaluation can be divided into two different phases:

(i) **A Priori estimates :** This is a theoretical performance analysis of an algorithm. Efficiency of an algorithm is measured by assuming the external factors.

(ii) **A Posteriori testing :** This is called performance measurement. In this analysis, actual statistics like running time and required for the algorithm executions are collected.

5 MARKS

1. Write the pseudo code for selection sort algorithm. *[PTA-4]*

Ans. The selection sort is a simple sorting algorithm that improves on the performance of bubble sort by making only one exchange for every pass through the list.

Pseudo code :

(i) Start from the first element i.e., index-0, we search the smallest element in the array, and replace it with the element in the first position.

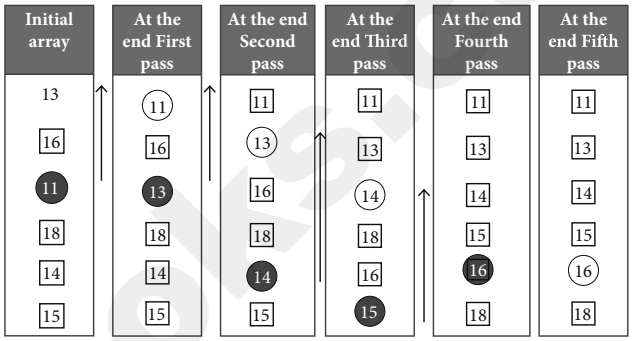
(ii) Now we move on to the second element position, and look for smallest element present in the sub-array, from starting index to till the last index of sub - array.

(iii) Now replace the second smallest identified in step-2 at the second position in the or original array, or also called first position in the sub array.

(iv) This is repeated, until the array is completely sorted.

Let's consider an array with values {13, 16, 11, 18, 14, 15}

Below, we have a pictorial representation of how selection sort will sort the given array.



In the first pass, the smallest element will be 11, so it will be placed at the first position.

(i) After that, next smallest element will be searched from an array. Now we will get 13 as the smallest, so it will be then placed at the second position.

(ii) Then leaving the first element, next smallest element will be searched, from the remaining elements. We will get 13 as the smallest, so it will be then placed at the second position.

(iii) Then leaving 11 and 13 because they are at the correct position, we will search for the next smallest element from the rest of the elements and put it at third position and keep doing this until array is sorted.

Government Exam Questions and Answers

1 MARK

1. What is another name for Binary search? *[QY-2019]*

(a) Linear (b) Half interval

(c) Decimal (d) Boolean

[Ans. (b) Half interval]

2. Which one of the following is not a factor to measure the execution time of an algorithm? *[Sep-2020]*

(a) Speed of the machine
(b) Operating system
(c) Programming language used
(d) Selection **[Ans. (d) Selection]**

5 MARKS

1. Explain the selection Sort Algorithm with an example. *[QY-2019]*

Ans. **(i)** Let us assume a list of n number of values stored in an array. Suppose if we want to search a particular element in this list, the algorithm that search the key element in the list among n elements, by comparing the key element with each element in the list sequentially.

(ii) The best case would be if the first element in the list matches with the key element to be searched in a list of elements. The efficiency in that case would be expressed as O(1) because only one comparison is enough.

(iii) Similarly, the worst case in this scenario would be if the complete list is searched and the element is found only at the end of the list or is not found in the list. The efficiency of an algorithm in that case would be expressed as O(n) because n comparisons required to complete the search.

(iv) The average case efficiency of an algorithm can be obtained by finding the average number of comparisons as given below:

Minimum number of comparisons = 1

Maximum number of comparisons = n

If the element not found then

maximum number of comparison = n

Therefore, average number

of comparisons = (n + 1)/2

(v) Hence the average case efficiency will be expressed as O (n).