



mca

VERSIONIZE

ISSUE 2021 | AUGUST



Articles

Solidity 101
Unix principles to Write Better Code
Time Management

Know The
Insights of Version'21

table of CONTENTS

1 About Version

Basic Introduction about the Version

2 Articles

Solidity 101, Unix Principles to Write Better Code

10 Time Management

Enables You to Work Smarter

13

Solution of Previous Edition

Crossword, 26-Skkido 1

14

Quiz

Where a Smart Answer Won't Get You Fired

19

Crossword

Improve Your Skills With Our Game

20

Spot the Difference

Let's Find Out Freedom from Boredom

21

Search In a Maze

Let's Take the Challenges

22

Version'21 Team

Committee Members

32 Solutions

Crossword, Spot the Difference, Search in a Maze

35 Versionize'21 Team

Magazine Team



VERSION, an annual All India MCA meet organized by the students of MCA, National Institute of Technology, Tiruchirappalli is one of the most renowned and remarkable All India MCA Technical Symposium.

Since its conception in 1991 Version is being organized successfully every year. It is a great honor to bring together all the students of MCA department from all over India under one roof of NIT-T where students showcase their talent, skills and potential. They fight to win, compete with enthusiasm and procure their precious knowledge. Version embraces the idea that inspire the students to think out of the box and expand their creative and mental ability.

Over the years VERSION was conducted within the campus. This year, we are organizing the 29th edition of VERSION. It is going to be in Virtual Mode i.e., online mode. We are putting in a lot of efforts to bring you the best platform and with your participation it will be a grand success. It is going to be a breathtaking showcase of the talent. With the current situation, conducting Version online could be a blessing in disguise, with a greater number of participants and hard work of us will make this year's VERSION, more competitive and fun. We are sure that you all will come across some astonishing and innovative events this year.

SOLIDITY 101

Solidity is one of the main programming languages used for writing smart contracts for the Ethereum blockchain. It is a high-level programming language that may look a lot like JavaScript, Python, and C++ in a glance.



Solidity is one of the main programming languages used for writing smart contracts for the Ethereum blockchain. It is a high-level programming language that may look a lot like JavaScript, Python, and C++ in a glance.

Solidity provides features such as inheritance, libraries and being statically typed, which makes it a very effective tool to build industrial order blockchain applications.



Solidity is a contract-oriented language, which means that smart contracts are responsible for storing all the programming logic that transacts with the blockchain.

The term smart in smart contracts is as the contract gets executed depending upon some logical condition and all nodes in blockchain have the same copy of that smart contract.

The EVM does not organically understand Solidity code; rather, Solidity ships with its own compiler that translates human-readable Solidity source code into low-level EVM-compatible bytecode.

CONTRACT STRUCTURE

Contracts in Solidity are like classes in object-oriented languages. Each contract can contain declarations of State Variables, Functions, Function Modifiers, Events, Structs Types and Enum Types.

State Variables

State variables are values which are permanently stored in contract storage.

```
contract SimpleStorage {  
    uint storedData; // State variable  
    // ...  
}
```

The following types are also called value types because variables of these types will always be passed by value, i.e., they are always copied when they are used as function arguments or in assignments.

Type	Keyword
Booleans	bool
Integers	int/uint
Address	address

Functions

Functions are the executable units of code within a contract.

```
contract SimpleAuction {  
    function bid() payable { // Function  
        // ...  
    }  
}
```

Events

Events are convenience interfaces with the EVM logging facilities.

```
contract SimpleAuction {  
    event HighestBidIncreased(address bidder, uint  
amount); // Event  
  
    function bid() payable {  
        // ...  
        HighestBidIncreased(msg.sender, msg.value); //  
Triggering event  
    }  
}
```

Structs Types

Structs are custom defined types that can group several variables.

```
contract Ballot {  
    struct Voter { // Struct  
        uint weight;  
        bool voted;  
        address delegate;  
        uint vote;  
    }  
}
```

Enum Types

Enums can be used to create custom types with a finite set of values.

```
contract Purchase {  
    enum State { Created, Locked, Inactive } // Enum  
}
```

A SMART CONTRACT FOR CELL PHONE

```
pragma solidity >=0.4.22 <0.9.0;  
  
contract CellSubscription {  
    uint256 monthlyCost;  
  
    constructor(uint256 cost) public {  
        monthlyCost = cost;  
    }  
  
    function makePayment() payable public {  
    }  
  
    function withdrawBalance() public {  
        msg.sender.transfer(address(this).balance);  
    }  
  
    function isBalanceCurrent(uint256 monthsElapsed) public  
view returns (bool) {  
        return monthlyCost * monthsElapsed ==  
address(this).balance;  
    }  
}
```

- This smart contract deals with the monthly subscription cost and has functionalities of balance withdrawn for payment of monthly subscription cost that is a subscriber facility.
- The subscriber pays a monthly fee in ether to the contract, and the cell phone company can check if the account is paid in full. Ethereum smart contracts act as wallets by default, meaning they can send, receive, and store ether just like a regular wallet address.
- **isBalanceCurrent** allows the cell phone company to check the status of the account on a given date.
- The public function called **withdrawBalance** that allows an account to be emptied.
- The payable function modifier tells the contract to automatically store any ETH internally that is sent when calling the **makePayment** function, which we will use later to determine if the subscriber has paid enough ether as of a given date.
- Notice how the **makePayment** function does not have a body at all; the storing of sent ether is automatic when the payable modifier is present.
- If desired, complex validation logic could be added to the function body, for example to reject partial payment.

Unix Principles → to Write Better Code

This article is about some of the Unix Principles or Philosophy of writing better and efficient code that we found relevant to be used in today's programming world, which should be applied in practice while writing modular codebases:



1 SIMPLE IS BETTER THAN COMPLEX

This is probably applicable in all domains in whichever we work, but this is mostly the most important principle that good programmers follow. If we don't take decisive actions while designing and writing our software, we will surely harvest complexity!

2 SMALL IS BEAUTIFUL

Going small reduces complexity, improves maintainability and testability. So rather than writing big monolithic code blocks, we should write small functions and try to design the interaction between those small functions efficiently.

3 MAKE EACH PROGRAM DO ONE THING WELL

This prevents redundant code in our software and improved readability. The more functionality we add to a program, the worse it gets!

4 BUILD A PROTOTYPE AS SOON AS POSSIBLE

Fail early, fail often, fail forward! We can only do this if we don't spend a vast amount of resources on each failure. To build a Minimum Viable Product (MVP) as soon as possible to avoid getting stuck in perfectionism.

5 CHOOSE PORTABILITY OVER EFFICIENCY

Portability is the ability of a system or a program to be moved from one environment to another and still function properly. While portability is an advantage, it comes at a cost: efficiency. You can reach very high degrees of efficiency by tailoring the software to one type of environment. However, the Unix philosophy advocates choosing portability over efficiency. The reason is simple: millions of users will work with the operating system.

6 USE SOFTWARE LEVERAGE TO YOUR ADVANTAGE

The first source of leverage for programmers is to tap into the collective wisdom of generations of coders before you. Use libraries rather than reinventing the wheel. Use StackOverflow and the wisdom of the crowd to find out how to fix bugs in your code. Talk to other programmers and ask them to review your code to find inefficiencies and bugs. All of those forms of leverage allow you to accomplish far more with less effort—more than you could ever accomplish alone!

7 CLARITY IS BETTER THAN CLEVERNESS

This principle specifically highlights the trade-off between clean and clever code—of course, it's great to write clever code, but it should generally not come at the costs of introducing unnecessary complexity.



⁸ ➤ DESIGN PROGRAMS TO BE CONNECTED WITH OTHER PROGRAMS

The rise of web services and microservices came from the willingness to share code and build on each other's code. Society benefits tremendously from open codebases and open interfaces because it reduces friction and investment overhead of all future code projects in the decades to come.

Your programs do not live in isolation. That's why you need to design the API (application programming interface) in a suitable way. If you write code with this principle in mind, you'll automatically design programs to be connected with other programs rather than programs that live in isolation.



⁹ ➤ MAKE YOUR CODE ROBUST

You'd call a thing robust—or a code base for that matter—if you cannot easily break it. You can accomplish robustness of your codebase by carefully designing access rights so that individual developers are not able to destroy the application without being forced to convince at least one additional person that the change is more likely to create than destroy value—yes, it comes at a price of agility but if you're not a one-person startup this price is worth paying.

TIME MANAGEMENT?

Is it easy to manage the TIME in this online mode? I can say partially YES and partially NO. Yes, because of course we all are sitting at our homes and doing work, and that's why it's flexible to change our schedule. It's easy to move our assigned task to a different time slot.

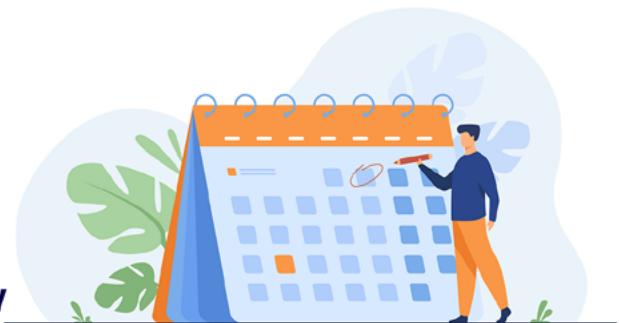


We all are NITians sitting at homes, doing assignments, projects, studies, any time according to our will without any boundaries. At the same time, the answer would be no, because of the variety of work we have to do. Also, some disturbance, or we can say divergence while we work, makes it challenging to manage time. Everyone is facing issues while it comes to Time Management, and it has now become a simple question "How to manage Time?"

We understand the value of time these days when we run a minute late in uploading our answer sheets and fear losing marks. However, no one can tell that "I manage time properly.". There is a need to use time effectively because each of us has the same 24 hours. This can be done if we follow some basic things, such as setting goals, making a schedule, removing distractions.

MAKING A SCHEDULE

Making a schedule always does not mean that you have to make a timetable from waking up until you sleep. It is just about writing your tasks on a simple sheet of paper, which has to be completed by deadlines. Then prioritize things based on the requirement of work or on what you love to do. And slowly and gradually follow that. You could write your assignments, projects, some critical work with deadlines and allot some time to each task.



SETTING GOALS

It's essential to decide your end goal to know whether you are providing your time to the right thing or not. Set minimal goals like for few hours and then again select something new to do. This method may be beneficial because if anything that was not in your schedule comes in between, you have small goals to complete, and then you can add something new.



TAKE TIME TO THINK

At first, it is necessary to think before allotting time to any task. Know the importance and its need before you give your time to it. Make a priority list according to the deadline or the necessity of work. Then, provide a suitable time slot and hours for the job with more value.



REMOVE DISTRACTIONS

Remove unwanted things or tasks with little importance. For example, the job which would not affect you, whether you do that or not, can be removed, or less time can be given to such things. It includes using your mobile phone or laptop for unnecessary things, which can either be avoided or done later when you are free from assigned tasks.



RELAX AND TAKE A BREAK

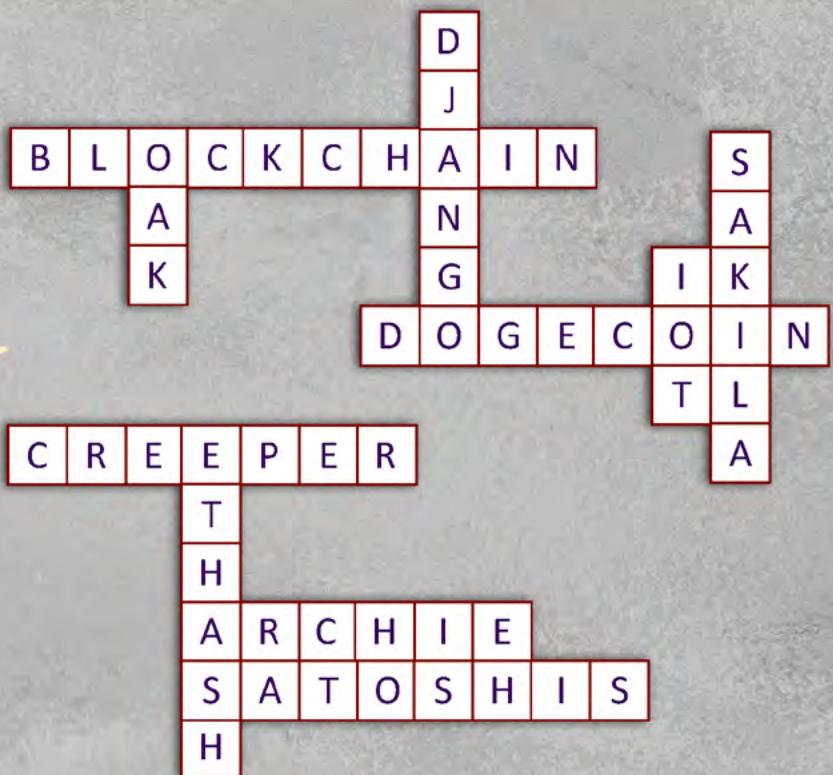
If you find difficulty in managing time, that's completely fine because no one could ever use time effectively. Everyone faces difficulty. So, sit and relax, take a break, meditate for a while and again start your work. It will help you in removing stress and managing your time more comfortably in the right direction.



We all know that there are many things while we are at home and with us, especially classes, assignments, meets, sitting for hours in front of the laptop. It is not as easy as it seems, but we could do it together and better if we start working on it. Charles Buxton said, "You will never find time for anything. If you want time, you must make it. " This is true. We can make out time for everything we need to do. The need is to use your time effectively and wisely.

SOLUTION OF PREVIOUS EDITION

CROSSWORD



F	A	L	S	E	A	W	A	I	T	R	U	E
P	N	O	N	L	O	C	A	L	I	A	S	P
I	F	E	L	S	E	X	C	E	P	T	I	O
C	O	L	L	E	C	T	I	O	N	S	Y	N
L	O	C	L	E	M	B	D	A	S	S	E	R
L	O	O	P	A	N	D	A	S	S	E	R	T
P	A	S	S	A	A	S	S	E	R	T	R	Y
E	X	C	E	P	T	K	I	N	T	E	R	L
R	A	I	S	E	I	M	P	O	R	T	L	E
G	L	O	B	L	E	F	I	N	A	L	L	Y
W	H	I	L	E	R	A	I	S	E	F	O	R
B	R	E	A	K	E	Y	I	E	L	D	O	G
W	I	T	H	I	M	G	H	D	R	I	V	E
F	T	U	R	T	L	E	L	E	M	E	N	T
P	A	S	S	N	D	H	D	R	A	I	S	E
F	O	R	T	E	M	P	F	A	I	L	D	M
A	S	Y	N	C	C	L	A	S	S	E	E	K
T	O	K	E	N	I	Z	E	B	R	O	N	I
F	O	R	A	S	Y	N	C	I	F	E	L	S
C	O	N	T	I	N	U	E	L	I	F	D	O
I	N	T	E	R	T	O	O	L	S	I	N	E
W	I	N	D	O	W	I	N	S	O	U	N	D
L	A	M	B	D	A	J	L	I	N	K	I	N
F	L	O	O	P	Y	B	R	A	I	N	L	Y
S	P	I	P	E	N	V	E	N	O	M	D	C
F	O	R	O	R	E	Q	U	S	T	S	S	S

26-SKKIDO 1

VERSION HISTORY TRIVIA QUIZ

1. What was the theme for 1st edition of VERSION? Also mention the year?

- (a) Interoperability , 1991
- (b) The Cerebral Autonomy , 1991
- (c) Satisfaction Always Resides With Perfection , 1991
- (d) The Cyber World , 1991

2. For which year the theme was “Security”?

- (a) 1999
- (b) 2001
- (c) 2002
- (d) 2000

3. What were the theme names for the years 1995 and 2009?

- (a) Continuing Challenges , Hyper Visor respectively
- (b) Continuing Challenges , Agile Computing respectively
- (c) The Cyber World , Hyper Visor respectively
- (d) Hyper Visor , Continuing Challenges respectively

VERSION HISTORY TRIVIA QUIZ

4. How many themes were there between “Journey Towards New Horizons” and “Capextremis”?

- (a) 23
- (b) 20
- (c) 21
- (d) 22

5. For which year the theme was “The Cyber World”?

- (a) 1994
- (b) 2006
- (c) 2020
- (d) None of the above

6. In which year the theme was not decided, and also, VERSION was not organized?

- (a) 1994
- (b) 2006
- (c) 2020
- (d) None

VERSION HISTORY TRIVIA QUIZ

7. In which year the theme was decided, but VERSION was not organized?

- (a) 1994
- (b) 2006
- (c) 2020
- (d) None

8. List all the themes that were decided for the year which is a multiple of 5.

9. With which theme(s) are you most familiar?

10. Name the themes that were of a single word.

VERSION HISTORY TRIVIA QUIZ

Answer

1. (c)
2. (b)
3. (a)
4. (d)
5. (a)
6. (b)
7. (c)
8. Continuing Challenges
Cybernetics
Interoperability
SAAS
Whizcon
Capextremis
9. Answer in your own
10. Cybernetics, Security, Convergence, Interoperability, Reengineering, SAAS, Droidian, Iamicus, Personigo, Crewcite, Whizcon, Witaura, Visnoetic, Qubykon, Knotryx, Capextremis, Deconsentro.



mca



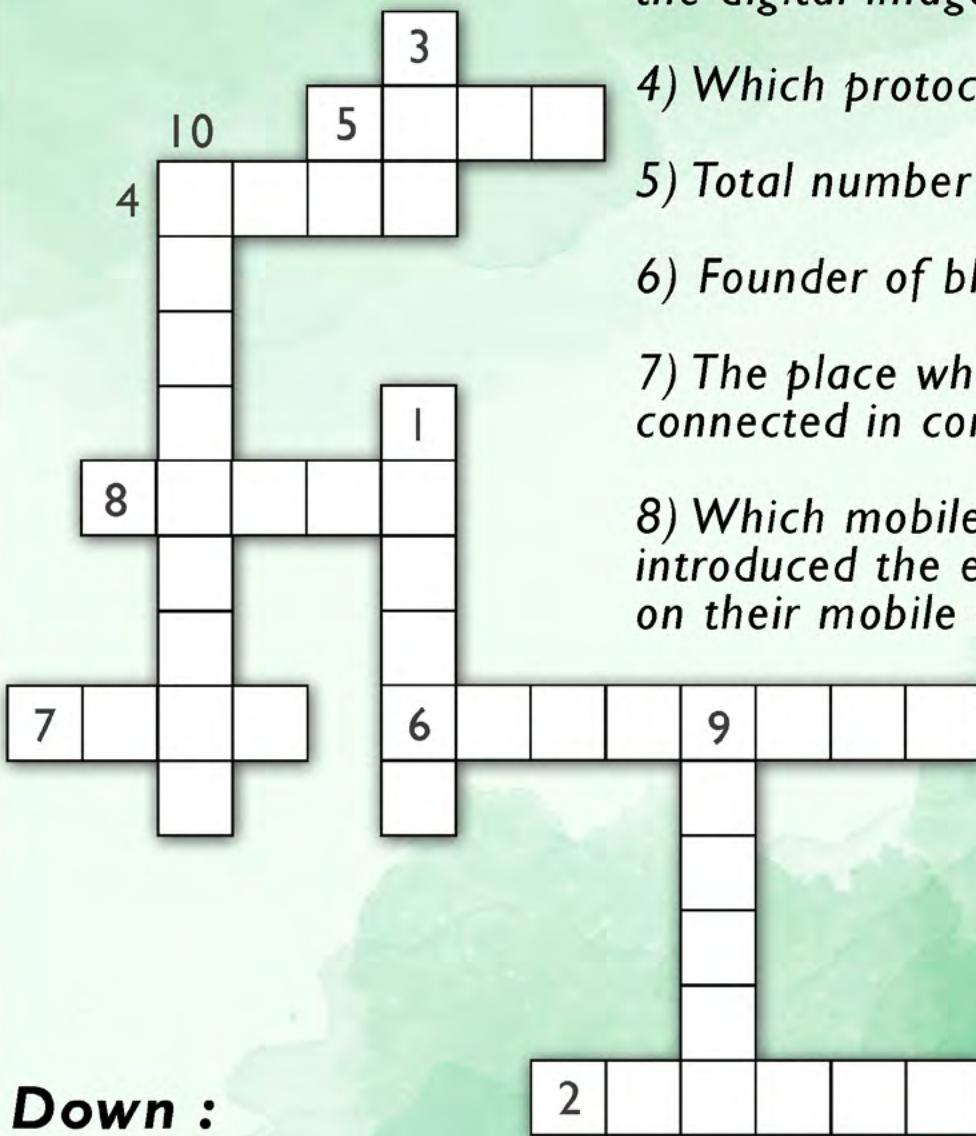
VERSIONIZE

GET EVERY STORY
RIGHT IN
YOUR HAND

www.version21.in

CROSSWORD

Across :



Down :

- 1) Which is the core and most important part of computer operating system which provide basic services for all part of os
- 3) Making of compressed digital archive might produce what type of file format
- 9) Which interface transmits one bit at a time
- 10) A type of synchronization tool

Spot The Differences

There are total 8 differences in the given picture. Find them in the bottom image.



Search in a maze

Given below is a maze having jumbled letters. Find all the eight words in the maze from list given below.

.CACHE

.ANALOGUE

.MALWARE

.ZIP

.BANDWIDTH

.ENCRYPTION

.FIREWALL

.PLUGIN

Z	D	B	A	L	Z	P	V	E	V	O	Q	A
A	B	C	Y	M	O	A	M	G	E	E	X	Z
W	T	Z	A	G	F	C	J	F	Y	N	P	L
S	U	N	Z	C	B	H	L	A	U	C	R	M
L	L	T	W	I	H	X	L	B	H	R	V	O
A	Y	A	J	M	D	E	A	N	B	Y	Z	P
Z	C	B	H	A	K	N	W	L	M	P	B	Q
O	I	O	J	L	D	C	E	D	F	T	F	S
L	P	Q	M	W	L	T	R	M	T	I	D	L
V	R	D	I	A	A	Z	I	P	K	O	M	I
E	S	D	U	R	N	G	F	T	L	N	R	L
F	T	P	G	E	A	R	Z	Y	Q	T	A	O
H	X	W	G	V	L	C	Y	U	V	H	J	S
V	R	X	L	Y	O	E	W	S	C	W	S	T
N	B	P	L	U	G	I	N	X	I	R	P	K
Q	B	O	A	D	U	R	Z	I	Y	A	H	P
Q	E	K	P	R	E	S	G	D	Q	T	H	U
C	F	O	D	F	S	M	S	N	J	P	P	J



Mayank Sharma
CHAIRMAN

Members



Bhumika Khakha
(Secretary)



Shivam Dubey
(Treasurer)



Akriti Upadhyay



Amandeep



Mahima Pawar



Himanshu Ghorwal



Naman Tripathi



Shivangi Gupta



Kuldeep Patidar
CHAIRMAN

Members



Jasbir



Anshu Mala Lakra



Saurabh Anand



Nikita Das



Amitabha Roy



Manish Kumar Pandey



Simran Garg



Hrishikant Mehta



Manish Nangliya



Aayushi



Shanoor Ahmed



Sachin Lalwani



Himanshi Bansal



Dimple Gyanani



Sonu Gupta



Piyush Devda



Payal Soni



Charu Bajaj



Harsh Rai



Yasser Osman Khan



Shobhit Samaria



Suruchi Bajaj



Amit Kumar



Harshit Omar



Ankit Singh



Paras Tiwari



Deepeeka



Pankaj Sharma



Sachin Sharma

Members



Mahima Pandey
CHAIRMAN



Sudhanshu



Sangam Raja



Ruchi Panse



Pooja Yadav



Prashansa Geete



KM Sarita Chaudhary



Asish Chowdhury



Neetu Pandit



Amar Sarkar



Saloni



Aman Shrivansh



Jayshree Choudhary



Moh Azhar Hussain



Sarath Kumar



Kajal Nerkar



Rajni Kasotiya



Sagar Gupta



Anand Prakash
CHAIRMAN

Members



Bikky Sharma



Vandana Mandloi



Khushboo Khanchandani



Prateek Dubey



Alok Khalkho



Praveen Vishwakarma



Sheetal Ingle



Mayank Singh



Neetu Meena



Kunal Kumawat



Shyam Shukla



Shivam Singh



Shubham Brajwasi



Nitesh Agrahari



Shubham Sharma



Dvarkesh Gupta

Members



Surbhi Dabhade
CHAIRMAN



Puneet Kumar



Rishikant Patel



Khushbu Mourya



Shashank Pandey



Nitin Dayar



Kishan Dwivedi



Aditya Soni



Akshay Gawai



Nitanshu Jain



Aakash Sastiya



Nitesh Kumar



Prashant Shukla



Shababul Ali



Sanskars Dhanotiya



Neeraj Upadhyay Amit Kumar Maddheshiya Manish Kumar



Dipesh Mandloi



Priya Chowdhury



Amit Gupta
CHAIRMAN

Members



Abhilash Bhadoliya



Amit Dwivedi



Amit Patidar



Arshpreet Kaur



Harsh Sharma



Ashish Tripathi



Mohit Gupta



Nandani Shah



Navin Parmar



Navnita Kumari



Ramratan Sharma



Rahul Dhadkar



Samaresh Maity



Nitin Prajapati



Vasu Namdev



Kanchan Kumar Thakur



Neeraj

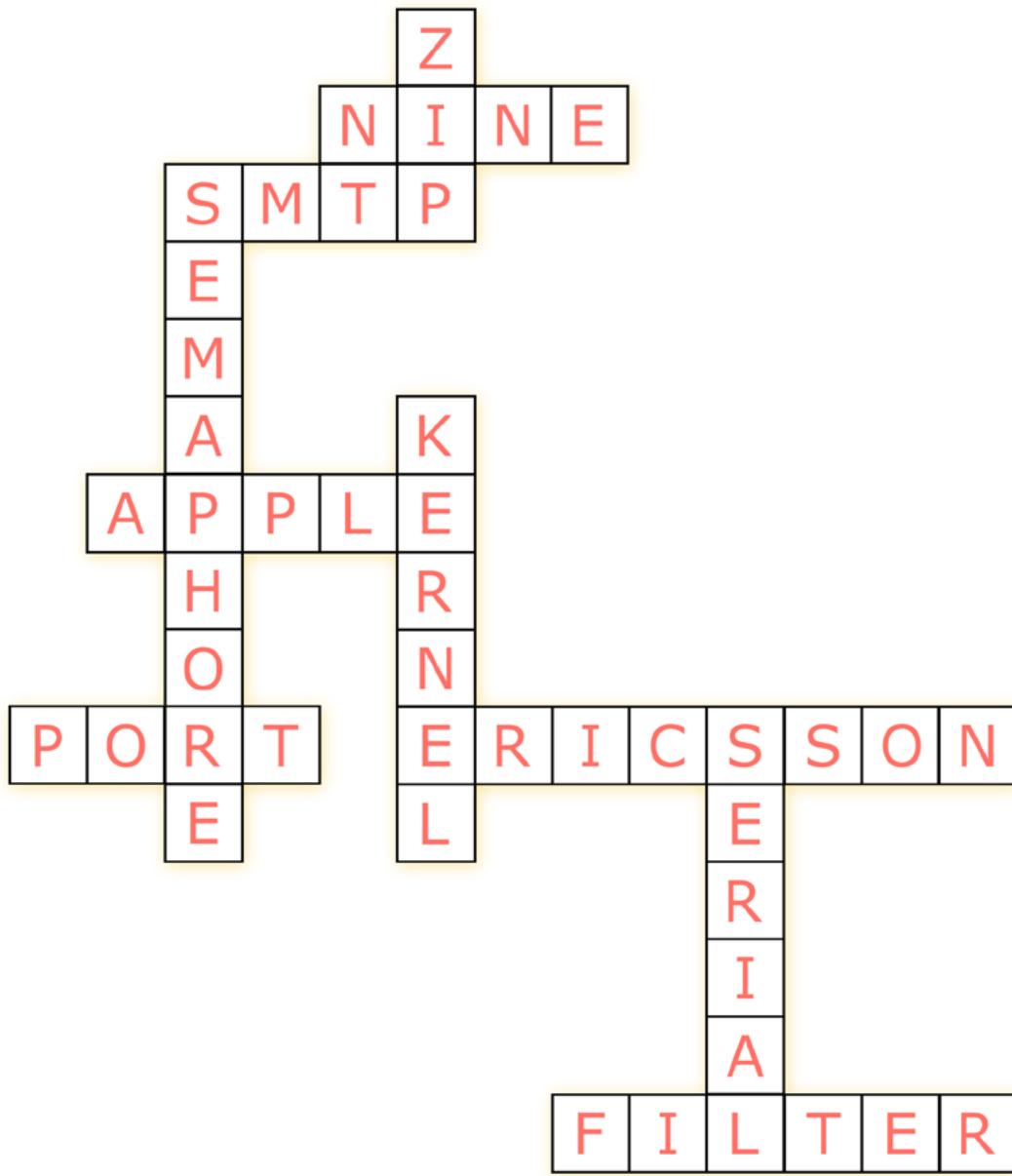


Uttam Kumar Sharma



Vishal Rai

CROSSWORD SOLUTION



Spot the Differences Solution



- ▼ 1. In the painting the circle is missing.
- ▼ 2. In another painting one triangle is in green color instead of yellow.
- 3. One leg of lamp is missing.
- 4. One wheel of rolling-chair is missing.
- 5. Some buildings are missing outside the mirror.
- 6. One thing is missing placed above the book on shelf.
- 7. One dot in clock is missing.
- 8. One ball is missing in the showpiece placed above book on table.

ANSWER- SEARCH IN A MAZE

Z	D	B	A	L	Z	P	V	E	V	O	Q	A
A	B	C	Y	M	O	A	M	G	E	E	X	Z
W	T	Z	A	G	F	C	J	F	Y	N	P	L
S	U	N	Z	C	B	H	L	A	U	C	R	M
L	L	T	W	I	H	X	L	B	H	R	V	O
A	Y	A	J	M	D	E	A	N	B	Y	Z	P
Z	C	B	H	A	K	N	W	L	M	P	B	Q
O	I	O	J	L	D	C	E	D	F	T	F	S
L	P	Q	M	W	L	T	R	M	T	I	D	L
V	R	D	I	A	A	Z	I	P	K	O	M	I
E	S	D	U	R	N	G	F	T	L	N	R	L
F	T	P	G	E	A	R	Z	Y	Q	T	A	O
H	X	W	G	V	L	C	Y	U	V	H	J	S
V	R	X	L	Y	O	E	W	S	C	W	S	T
N	B	P	L	U	G	I	N	X	I	R	P	K
Q	B	O	A	D	U	R	Z	I	Y	A	H	P
Q	E	K	P	R	E	S	G	D	Q	T	H	U
C	F	O	D	F	S	M	S	N	J	P	P	J



Anand Prakash
Editor

Design Team



Alok Khalkho



Bikky Sharma



Mayank Kumar Singh

Editorial Team



Mayank Sharma



Sagar Gupta



Paras Tiwari



Saloni



Asish Chowdhury



Sudhanshu



Kajal Nerkar



Prashansa Geete



Nitin Dayar



Akshay Gawai



Pooja Yadav



Aman Shrivansh



PUBLISHED BY

MCA, Department of Computer Applications
National Institute Of Technology, Tiruchirappalli



[f /versionmeet](#)

[o /version_nit_trichy](#)

[y /version_nit_trichy](#)