Programming Club

UNDER THE HOOD

END-TERM EVALUATION

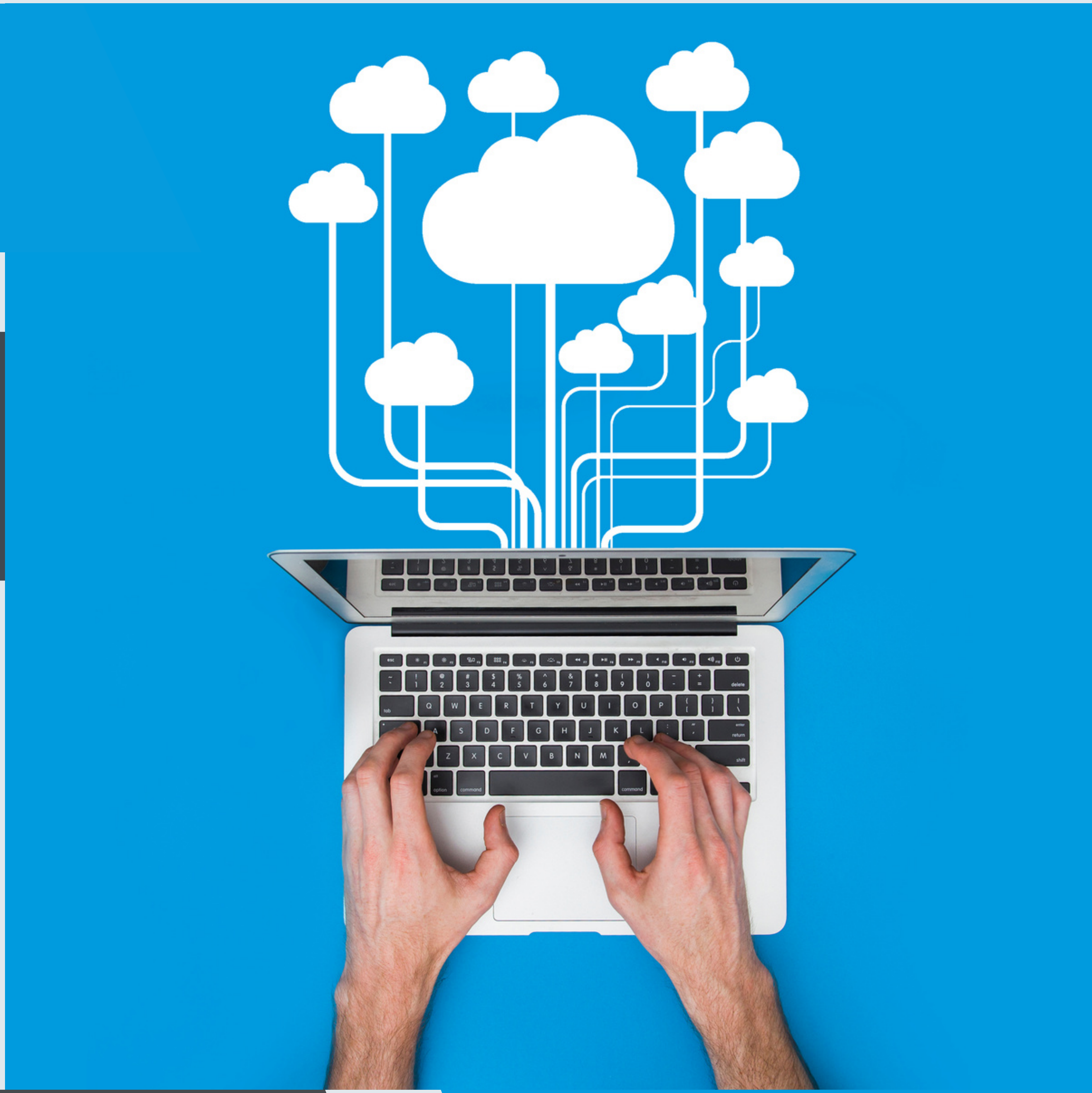


Table Of Content

Our Team

Timeline

Assembly Language

Projects

Our Team

Project mentors

Aditya Tanwar

Akhil Agrawal

Mentees

Y00010100

Ramtej Penta
Hisham Hadi T
Satender Kumar Yadav

VRAD

Dhruv
Abhir Rajbongshi
Rishi Poonia
Vishant Bhadana

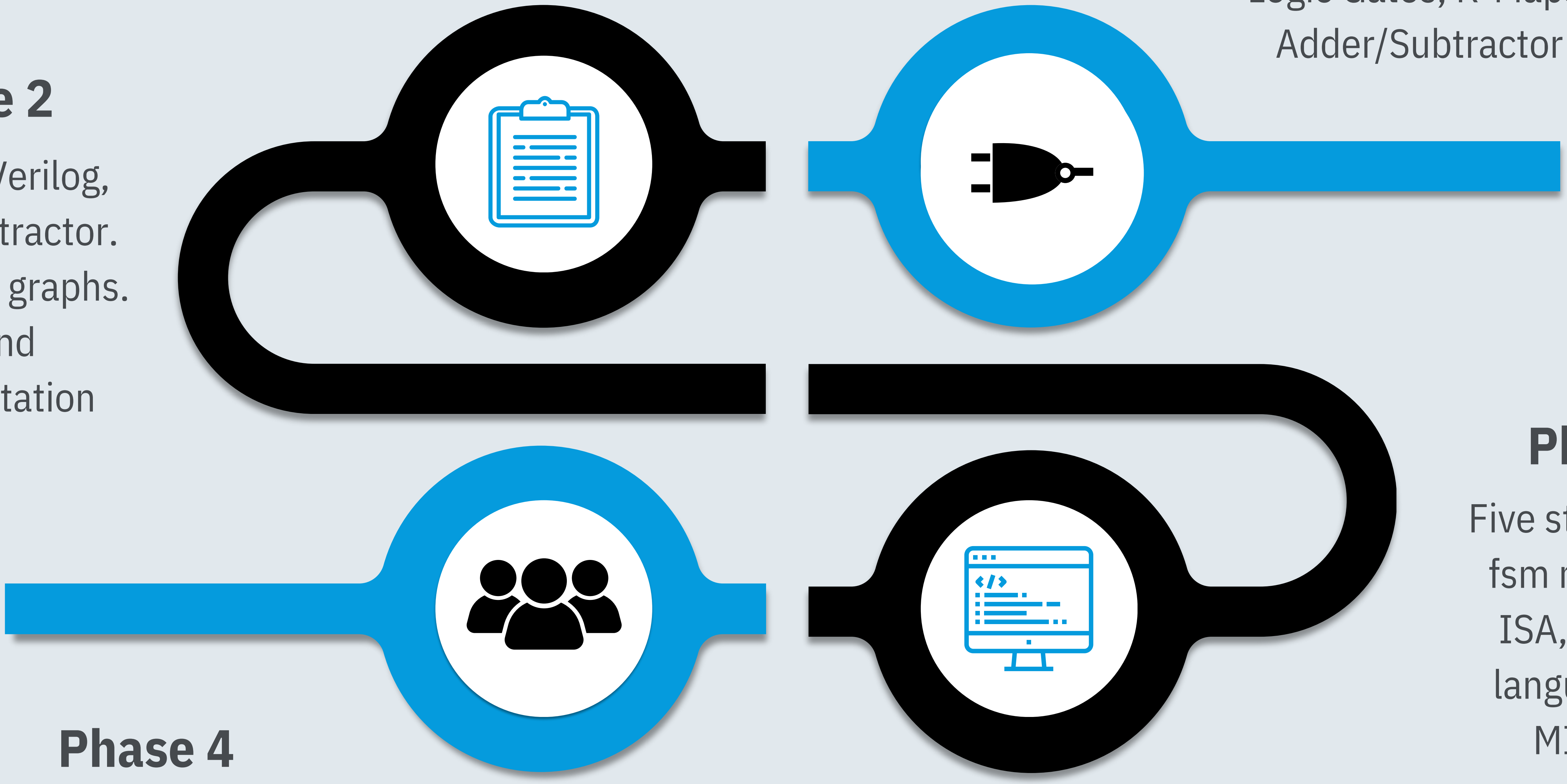
qtsimp

Anuj
Ujjawal Dubey
Lavesh Gupta
Rajat Gattani
Aayush Agrawal

Surviving with Google

Madhur Bansal
Suvrat Pal
Lokesh Nayak

Timeline



Phase 1

Basics of Binary numbers,
Logic Gates, K-Maps,
Adder/Subtractor

Phase 2

Basics of Verilog,
Adder/Subtractor.
Gtkwave for graphs.
FSM and
implementation

Phase 3

Five state abstract
fsm model of an
ISA, assembly
language using
MIPS ISA.

Phase 4

Implementation of
topics learnt
throughout the
project to make a
application in MIPS
as a part of team

ASSEMBLY

We learnt some of the concepts fundamental to any assmebly language, specifically in MIPS-

- The concept of registers to store memory address and simulate variables
- The rigidity of operands throughout (two/three registers as arguments only)
- Using stack to store values of variables and allow recursion
- Combined all these concepts to make programs for GCD and Fibonacci computation (both iterative and recursive)

PROJECTS

The final phase was implementation of everything learnt by making four projects with user friendly interface for which Mentees were divided into 4 teams with 3-5 members each.

Simple calculator

Date converter

Tic Tac Toe

Scientific calculator

Simple Calculator

Team - Surviving With Google

Brief Description

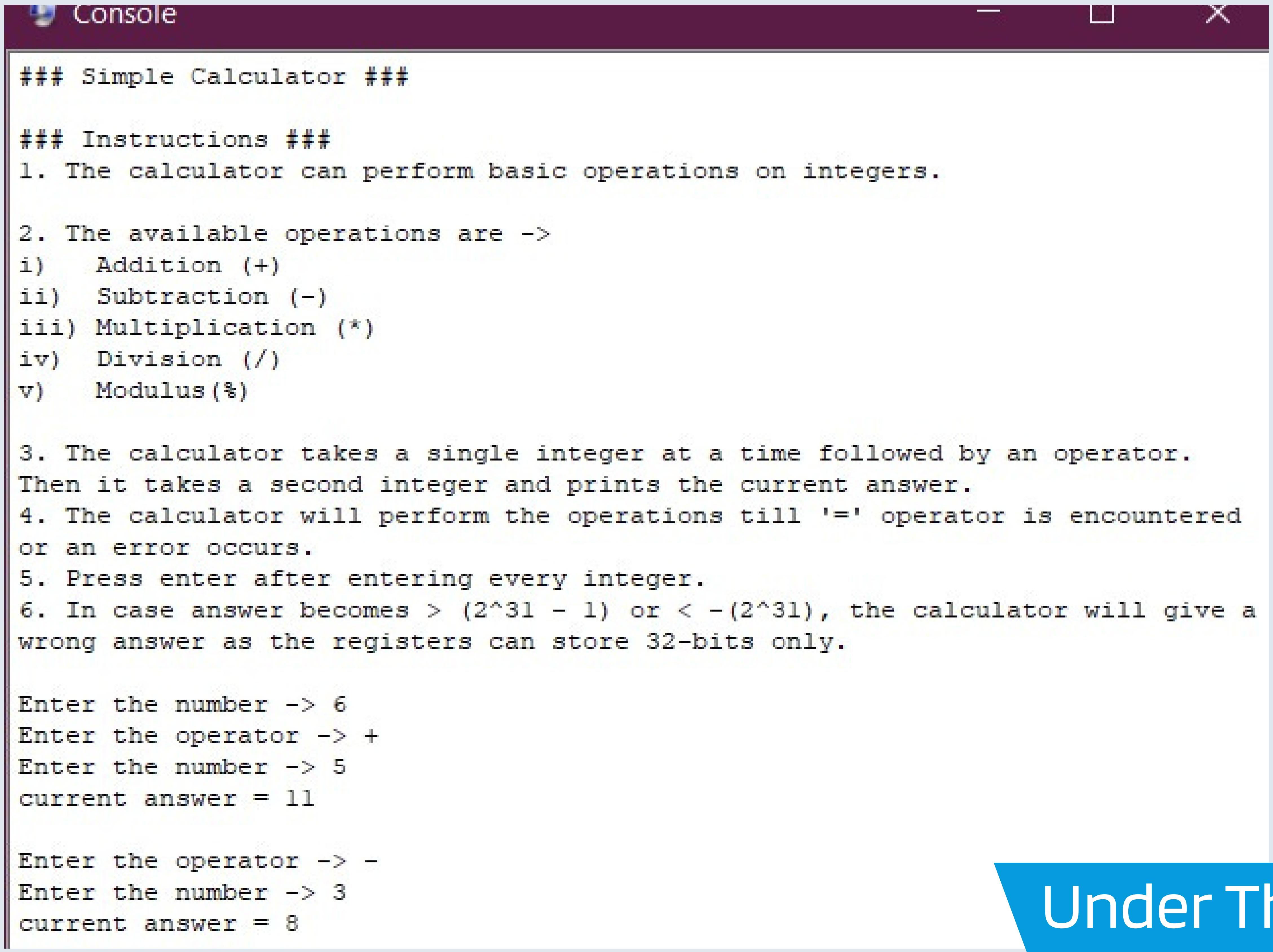
- Implementation of simple caculator with user friendly interface

- Calculator can perform arithmetic operations such as addition, subtraction, multiplication, division and modulus

- It will take a single input at a time, and output the answer at each step, hence evaluating the given expression from left to right.

Simple Calculator

User-Interface



```
### Simple Calculator ###

### Instructions ###
1. The calculator can perform basic operations on integers.

2. The available operations are ->
i)   Addition (+)
ii)  Subtraction (-)
iii) Multiplication (*)
iv)  Division (/)
v)   Modulus(%)

3. The calculator takes a single integer at a time followed by an operator.
Then it takes a second integer and prints the current answer.
4. The calculator will perform the operations till '=' operator is encountered
or an error occurs.
5. Press enter after entering every integer.
6. In case answer becomes > (2^31 - 1) or < -(2^31), the calculator will give a
wrong answer as the registers can store 32-bits only.

Enter the number -> 6
Enter the operator -> +
Enter the number -> 5
current answer = 11

Enter the operator -> -
Enter the number -> 3
current answer = 8
```

Date Converter

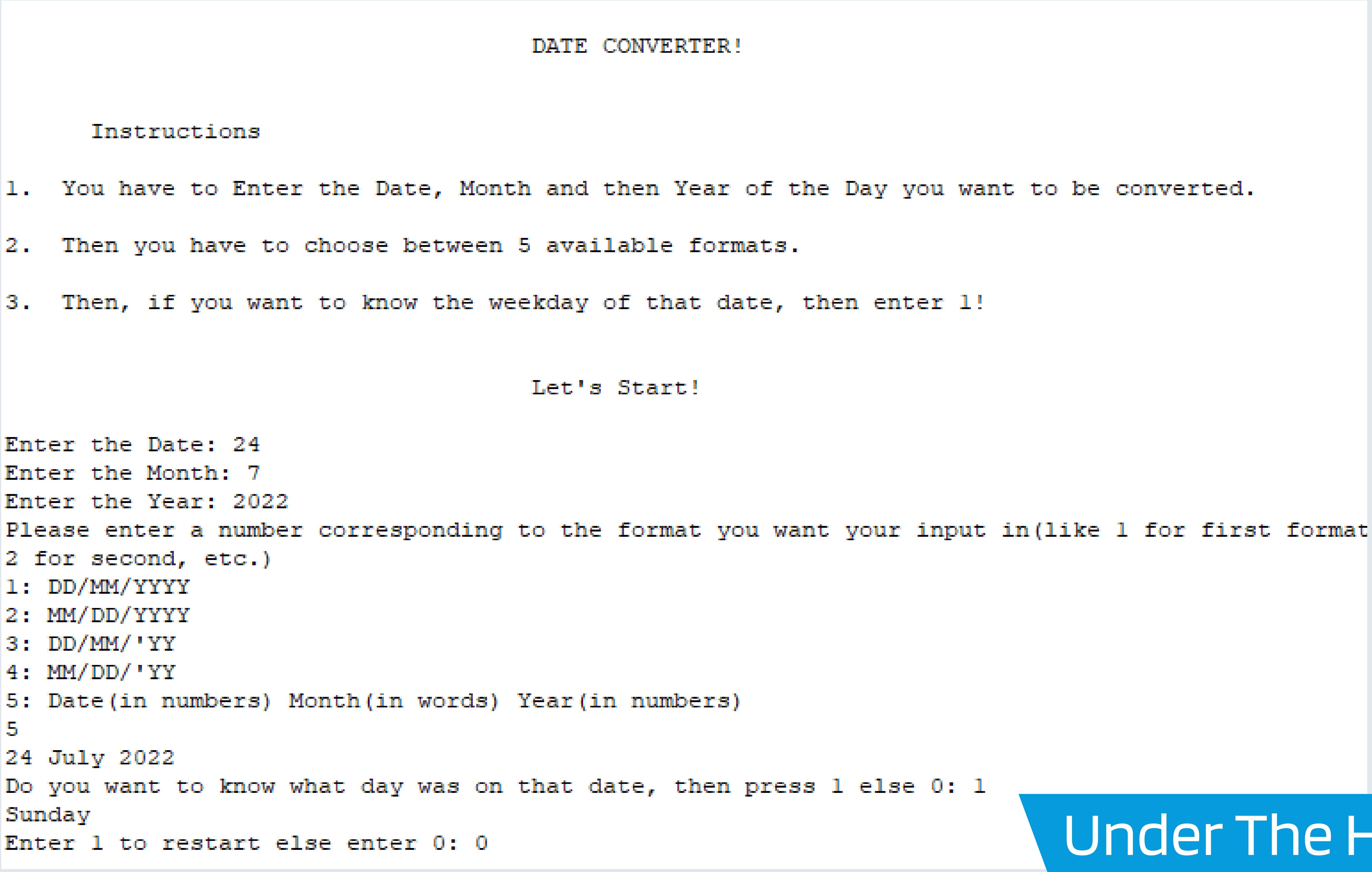
Team - VRAD

Brief Description

- Implementation of Date Converter with interface containing all instructions for users
- Takes input of date ,month ,year and gives options to user to convert the date in one of 5 different formats
- Finally a function to print weekday on that given date, restart or end the program

Date Converter

User-Interface



```
                                     DATE CONVERTER!


      Instructions

1.  You have to Enter the Date, Month and then Year of the Day you want to be converted.

2.  Then you have to choose between 5 available formats.

3.  Then, if you want to know the weekday of that date, then enter 1!


                                     Let's Start!

Enter the Date: 24
Enter the Month: 7
Enter the Year: 2022
Please enter a number corresponding to the format you want your input in(like 1 for first format
2 for second, etc.)
1: DD/MM/YYYY
2: MM/DD/YYYY
3: DD/MM/'YY
4: MM/DD/'YY
5: Date(in numbers) Month(in words) Year(in numbers)
5
24 July 2022
Do you want to know what day was on that date, then press 1 else 0: 1
Sunday
Enter 1 to restart else enter 0: 0
```

Tic-Tac-Toe

Team - qtSimp

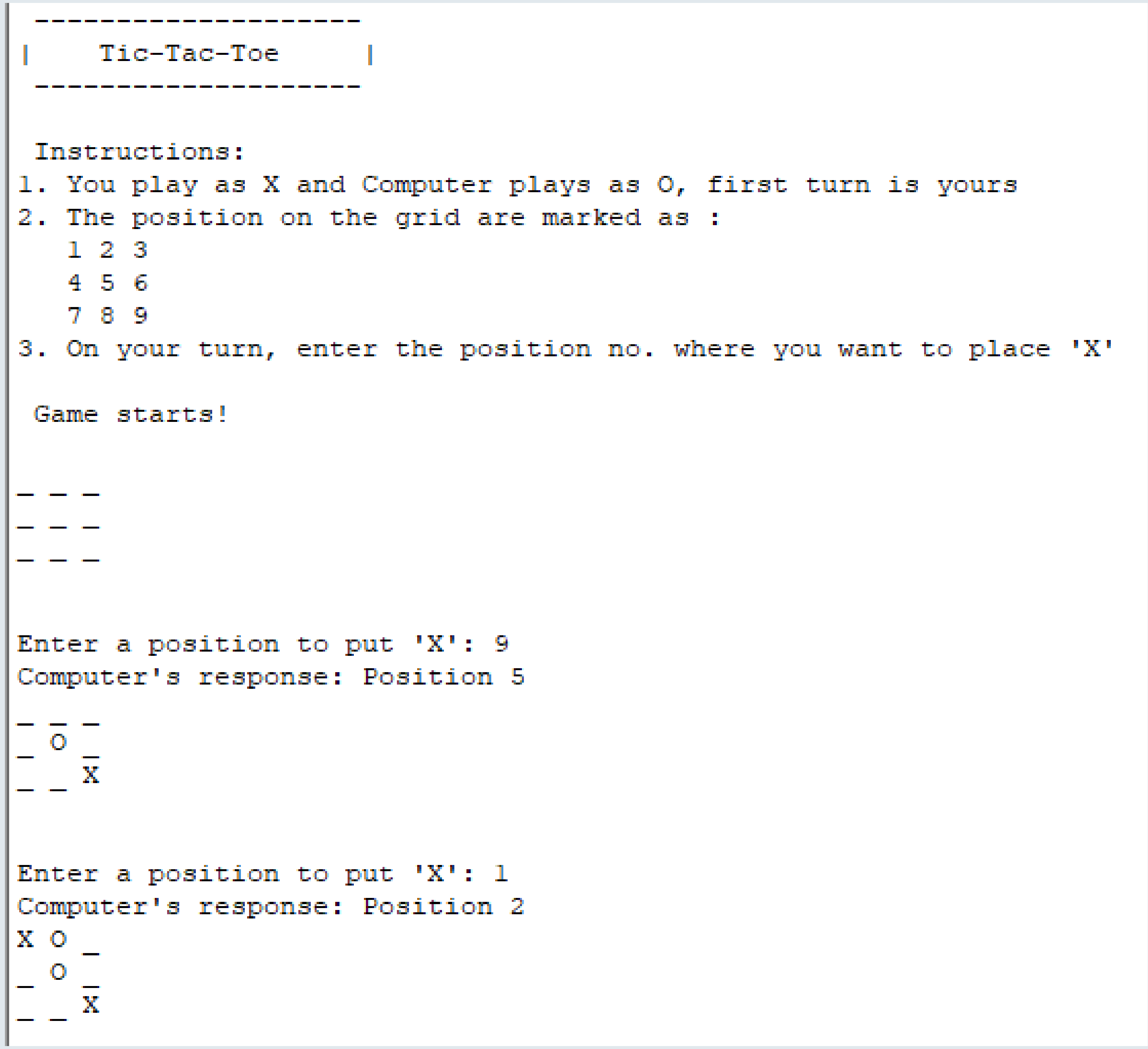
Brief Description

- Implementation of popular game tic-tac-toe with user friendly interface

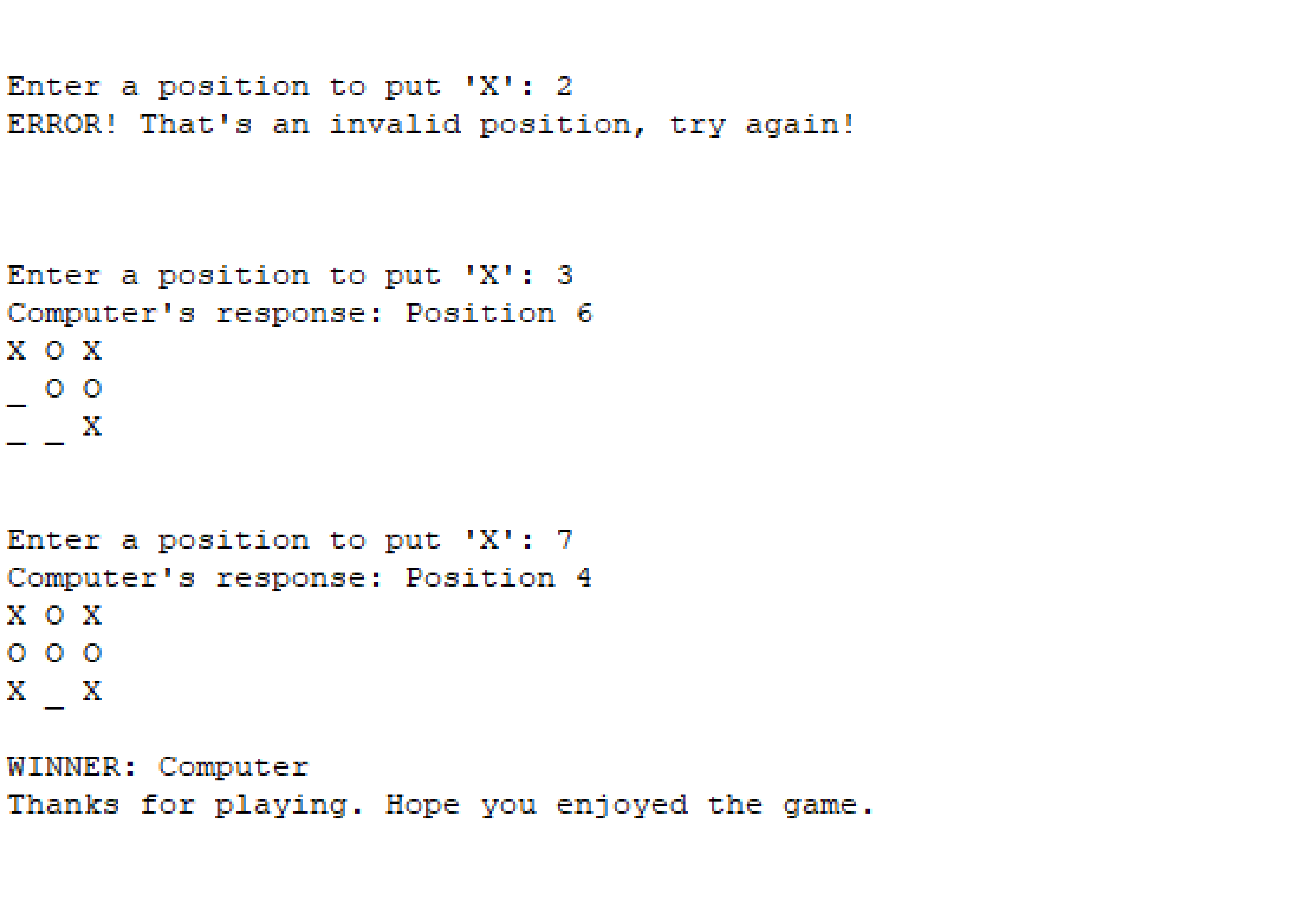
1. The computer presents the instructions to user
2. It prompts the user for a position to place 'X' at
3. Validates the user's input
4. Finds the optimal response to current game
5. Prints the updated grid
6. Checks for victory / draw

Tic-Tac-Toe

User-Interface



```
 --------------------
|    Tic-Tac-Toe     |
 --------------------

 Instructions:
1. You play as X and Computer plays as O, first turn is yours
2. The position on the grid are marked as :
   1 2 3
   4 5 6
   7 8 9
3. On your turn, enter the position no. where you want to place 'X'

 Game starts!


_ _ _
_ _ _
_ _ _


Enter a position to put 'X': 9
Computer's response: Position 5
_ _ _
_ O _
_ _ X


Enter a position to put 'X': 1
Computer's response: Position 2
X O _
_ O _
_ _ X
```



```
Enter a position to put 'X': 2
ERROR! That's an invalid position, try again!



Enter a position to put 'X': 3
Computer's response: Position 6
X O X
_ O O
_ _ X


Enter a position to put 'X': 7
Computer's response: Position 4
X O X
O O O
X _ X

WINNER: Computer
Thanks for playing. Hope you enjoyed the game.
```

Scientific Calculator

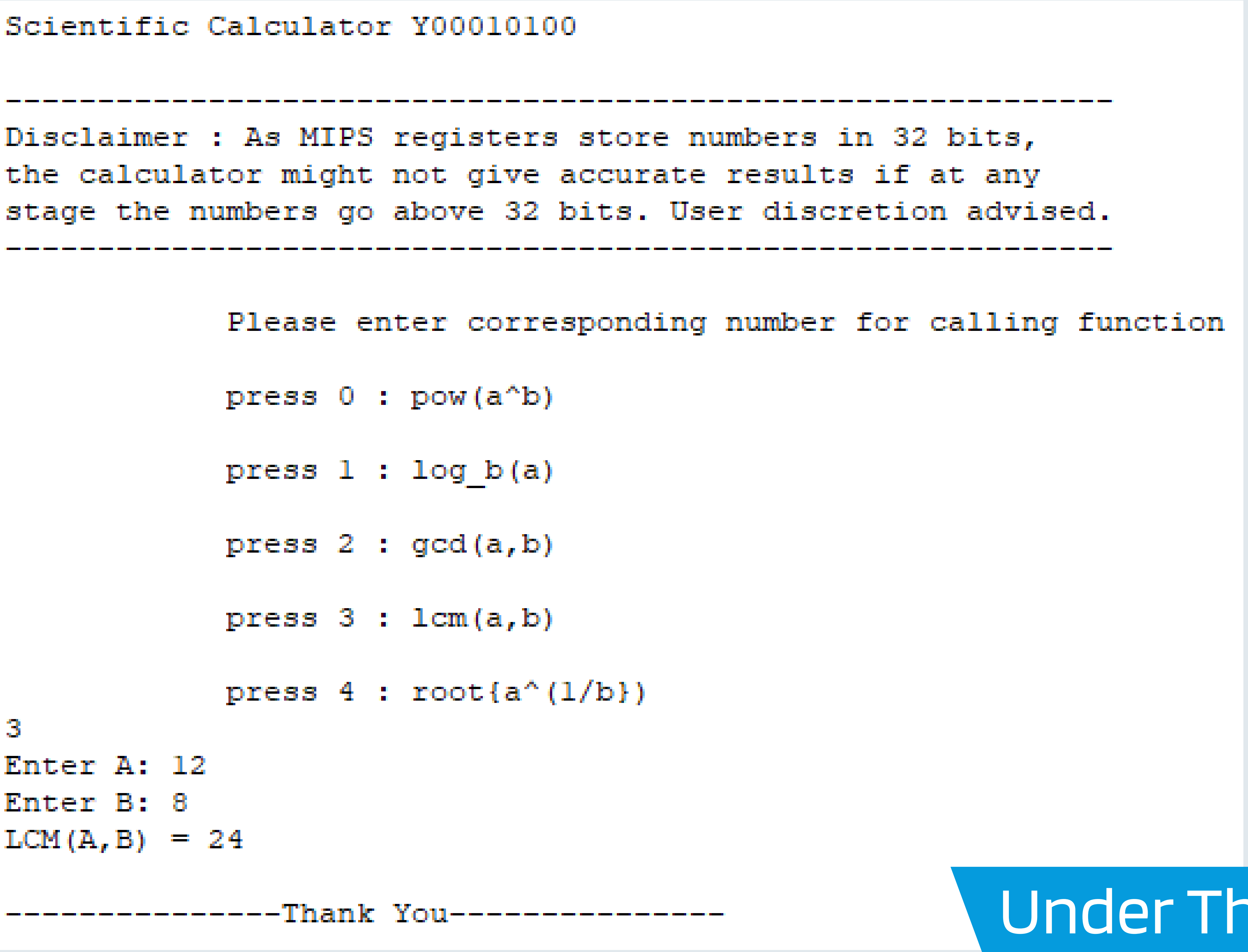
Team - Y00010100

Brief Description

- A Scientific calculator which can do the following operations on the two numbers a and b given in the input:
  1) a^b modulo 4999
  2) logarithm of "a" to the base "b"
  3) gcd of a,b
  4) lcm of a,b
  5) a power 1/b
  The interface shows all the instructions for the users. Users can refer to the interface and continue using the calculator.

Scientific Calculator

User-Interface



```
Scientific Calculator Y00010100

------------------------------------------------------------
Disclaimer : As MIPS registers store numbers in 32 bits,
the calculator might not give accurate results if at any
stage the numbers go above 32 bits. User discretion advised.
------------------------------------------------------------

            Please enter corresponding number for calling function

            press 0 : pow(a^b)

            press 1 : log_b(a)

            press 2 : gcd(a,b)

            press 3 : lcm(a,b)

            press 4 : root{a^(1/b})
3
Enter A: 12
Enter B: 8
LCM(A,B) = 24

---------------Thank You---------------
```

THANK YOU