# CC-112L

# Programming Fundamentals

# Laboratory 01

# Introduction to Programming, Algorithms and C

**Version: 1.0.0**

**Release Date: 14-07-2022**

**Department of Information Technology**

**University of the Punjab**

**Lahore, Pakistan**

## Contents:

# Background and Overview:

**What is a Program?**

A program is a sequence of instructions to perform a specific task. A computer program is set of instructions given to the computer to perform a particular task. A computer program in its human-readable form is called source code.

**What is an Algorithm?**

An algorithm is a finite set of instructions used to solve a particular task. Everything we do follows an algorithm. So, computer algorithms are instruction manuals that tells computer how to perform a task.

**Pseudocode:**

Pseudocode is a of an algorithm in human readable form that does not requires any programming language. It summarizes the flow in which program executes.

**Flowchart:**

A flowchart is a pictorial representation (diagram / visual representation) of an algorithm, or process in a sequential order.

**Microsoft ® Visual Studio:**

Microsoft ® Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

Microsoft ® Visual Studio supports 36 different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic, .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

© https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

**LARP:**

LARP is in an acronym. It is a compression of the phrase **"Logics of Algorithms and Resolution of Problems"**. LARP is a programming language for the purpose of rapid prototyping of algorithms.

LARP's main advantage over traditional programming languages is its flexible and semi natural syntax, allowing one to formulate algorithms without the impediments of cryptic languages such as C, C++, Pascal or Java.

© http://larp.marcolavoie.ca/en/description/description.htm

## Activities:

### Pre-Lab Activities:

**Microsoft ® Visual Studio 2022 Installation:**

- Open Web browser and in search bar, type **"Visual Studio",** Open the first site or
- Type the URL https://visualstudio.microsoft.com in the address bar of the Web browser



Fig. 01 (Microsoft ® Visual Studio Installation)
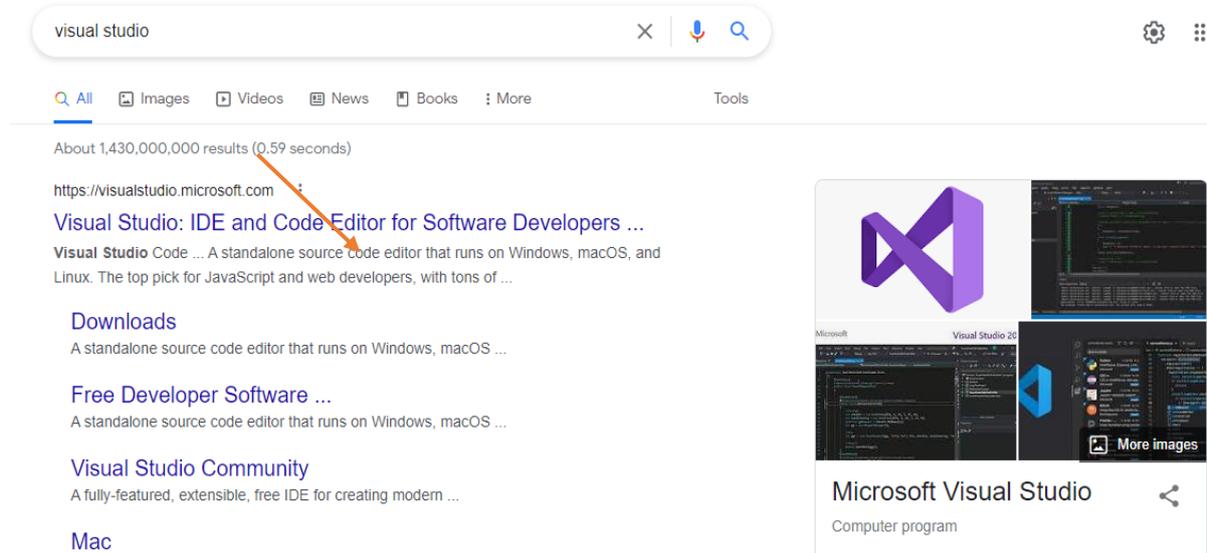


Fig. 02 (Microsoft ® Visual Studio Installation)

- Under **"Meet the Visual Studio family"** heading, click **"Download Visual Studio"** dropdown menu
- Click **"Community 2022".** The Visual Studio Setup will be downloaded.
- Run the **"Visual Studio Setup"** file. Visual Studio Installer will be downloaded.

Fig. 03 (Microsoft ® Visual Studio Installation)

- Launch **"Microsoft ® Visual Studio Installer"**
- In **"Workloads"** tab, under **"Desktop & Mobile"**, click the checkbox against **"Desktop development with C++"**
- On bottom right corner, click **"Install"**
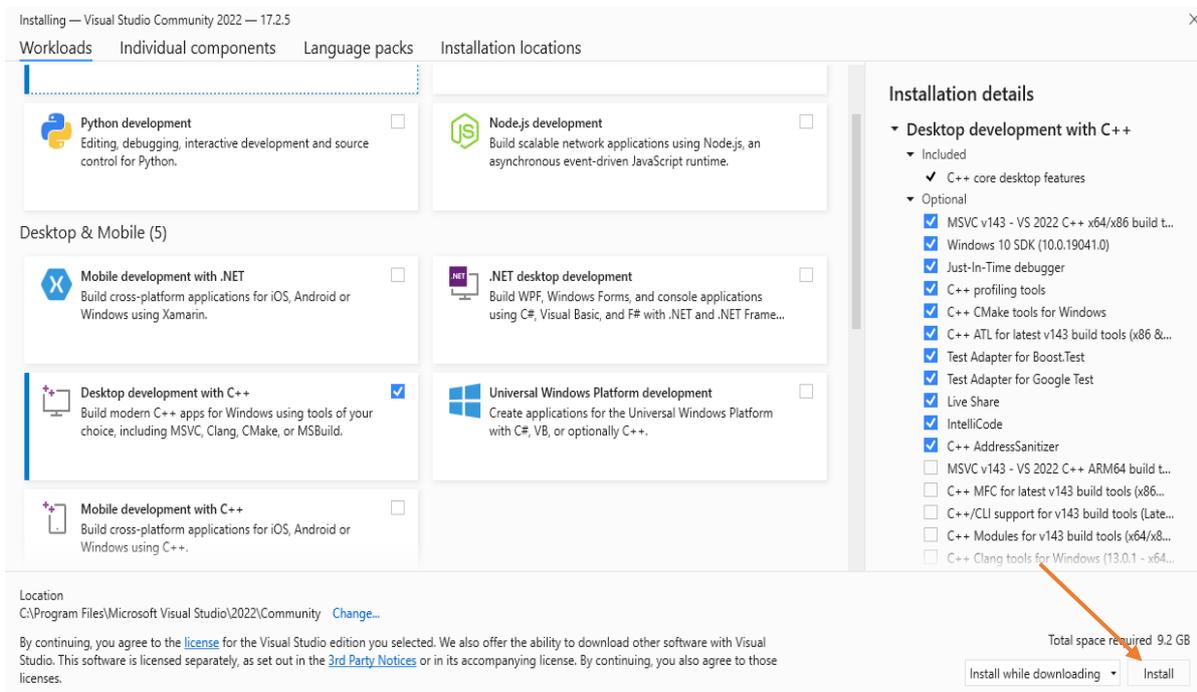


Fig. 04 (Microsoft ® Visual Studio Installation)

The **"Microsoft ® Visual Studio 2022"** installation starts.

**LARP (Logic of Algorithms for Resolution of Problems) Installation:**

- Open Web browser
- In search bar, type **"LARP Software".** Open the first site or
- Type the URL https://en.freedownloadmanager.org/Windows-PC/LARP.html  in the address bar of the Web browser
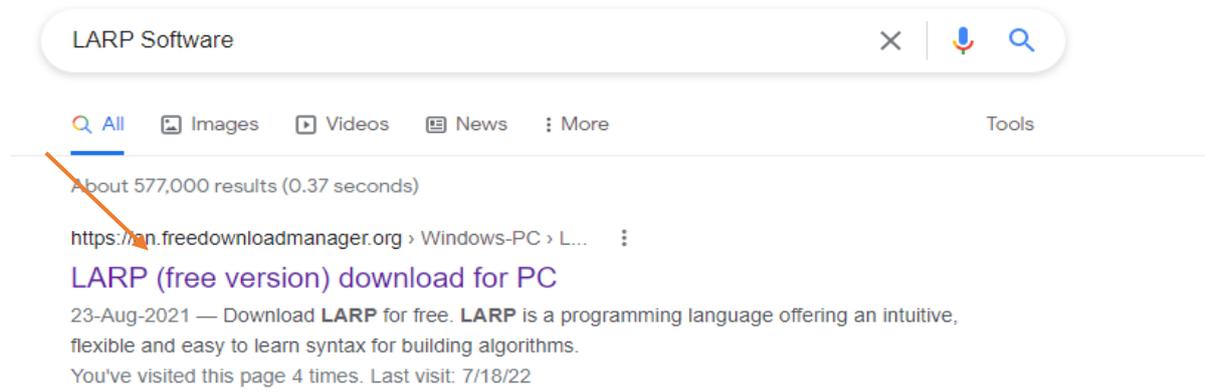
Fig. 05 (LARP Installation)

- Click **"Download now"**. LARP setup will be downloaded
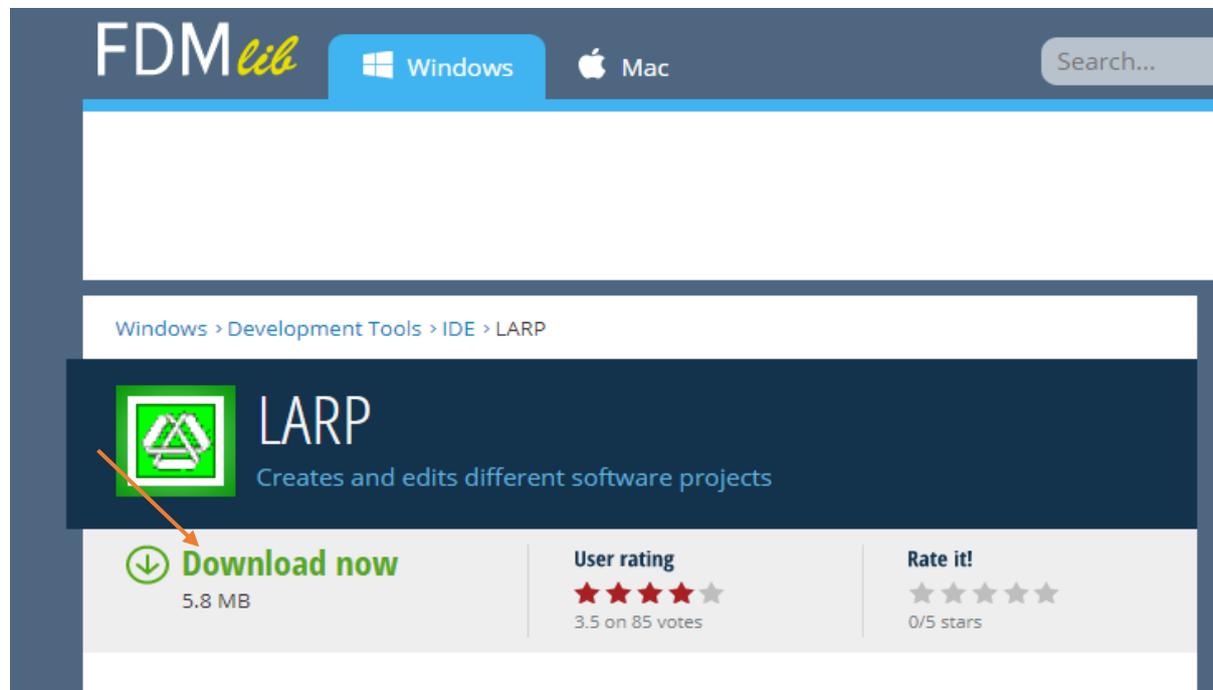


Fig. 06 (LARP Installation)

- Run the LARP setup
- Follow the Setup wizard instructions. LARP will be installed
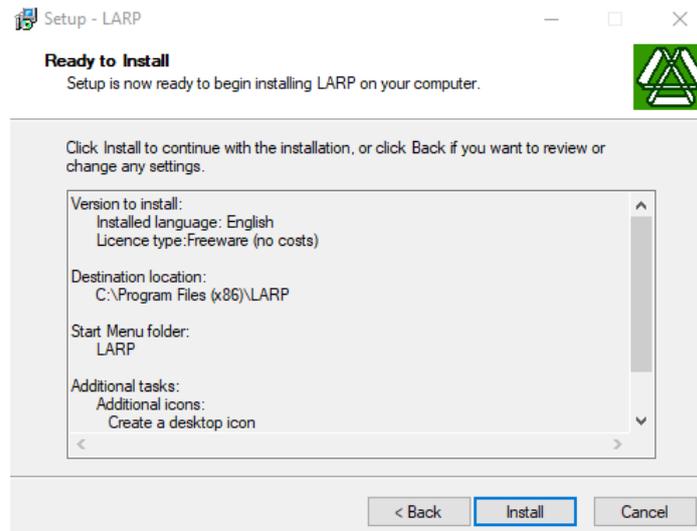
Fig. 07 (LARP Installation)

**In-Lab Activities:**

**Pseudocode**

**New Project:**

- Double Click **"LARP"** application on the Desktop
- Press **"CTRL + N"** on the keyboard OR on the navigation bar, click **"File"**, then click **"New"** from the dropdown
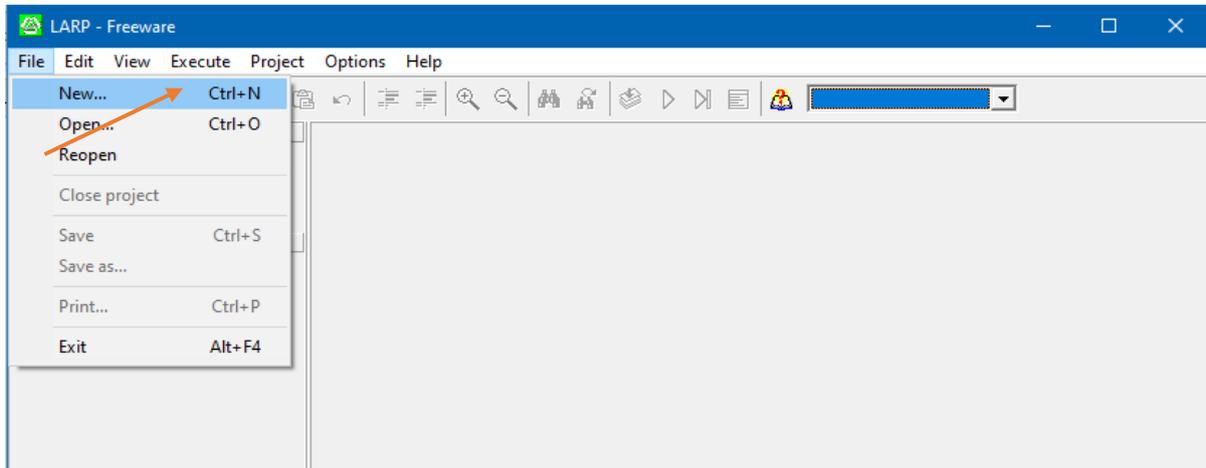


<p align="right">Fig. 08 (LARP New Project)</p>

**"New"** dialog box opens.

- Under **"New project"**, click **"Pseudo code"**
- Click **"OK"**
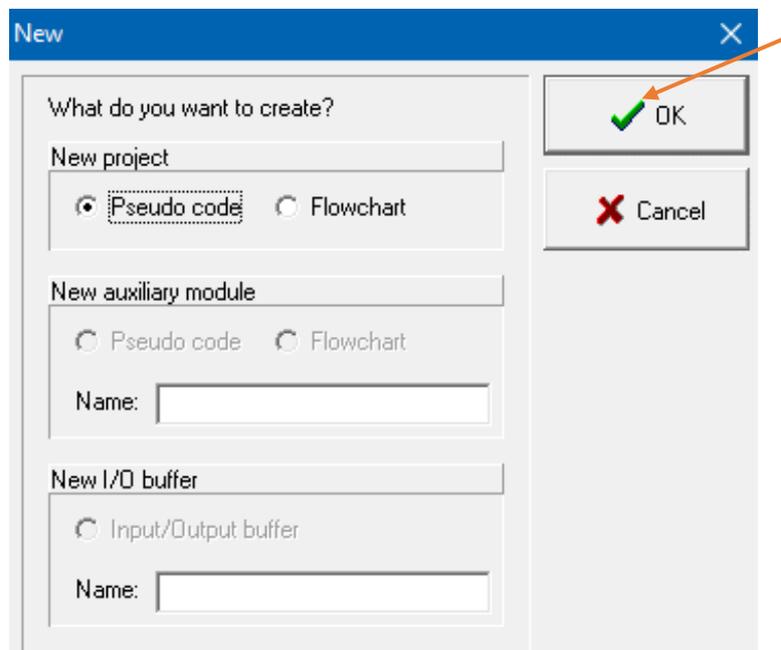


<p align="right">Fig. 09 (LARP New Project)</p>

New Pseudocode project will be created.
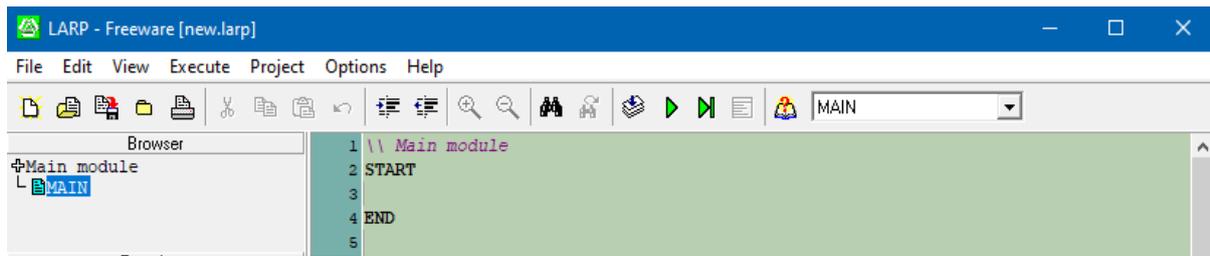
Fig. 10 (Pseudocode in LARP)

**Display Output on Console:**

- On the text editor in LARP, type **"WRITE {expression}"** OR drag the **"WRITE {expression_list}"** template from the **"Templates"** column
- Replace the expression list with your desired text within inverted commas
- On the ribbon, click the execute button OR press **"F7"** key
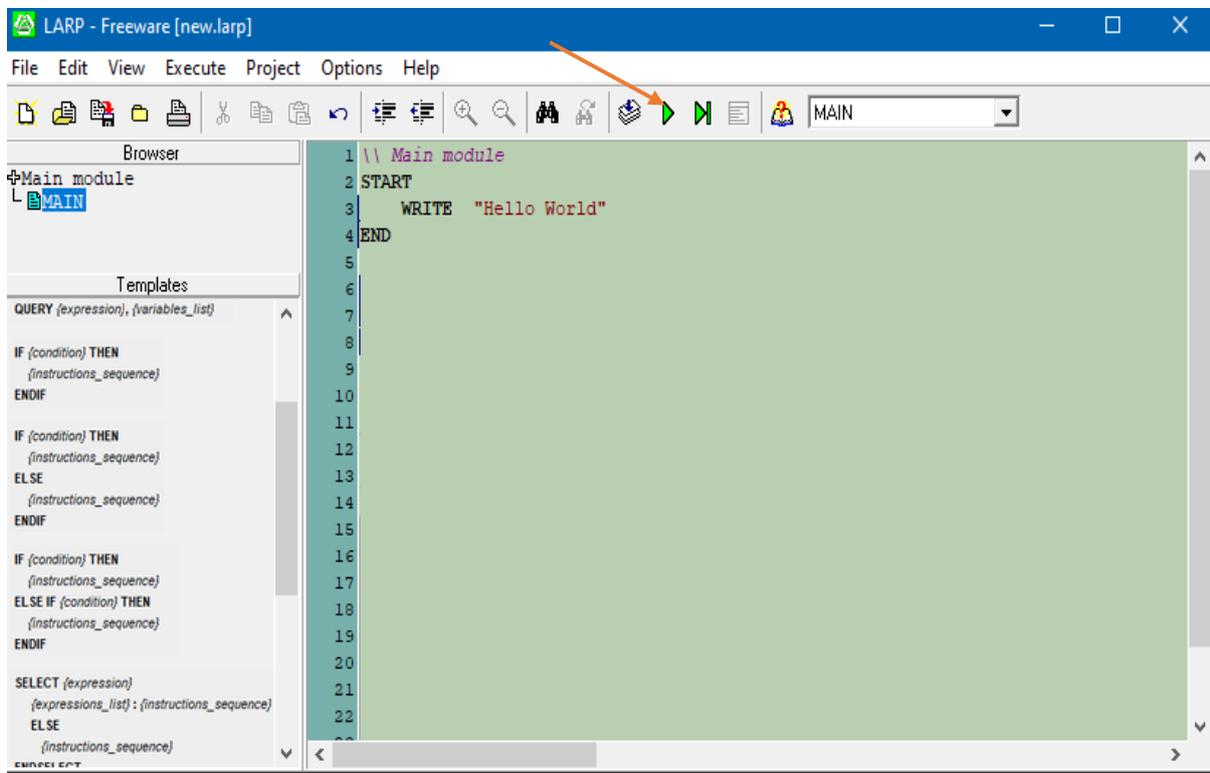


Fig. 11 (Pseudocode in LARP)

Desired output will be displayed on the console.
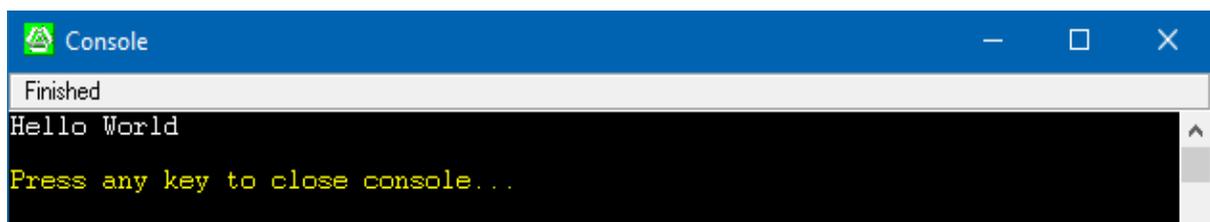


Fig. 12 (LARP Console)

**Read & Display Input on Console:**

- On the text editor in LARP, type **"READ {variables}"** OR drag the **"READ {variable_list}"** template from the **"Templates"** column
- Replace the variable list with your desired variable name
- On the next line type **"WRITE"** following with an expression and comma separated variable name
- On the ribbon, click the execute button OR press **"F7"** key
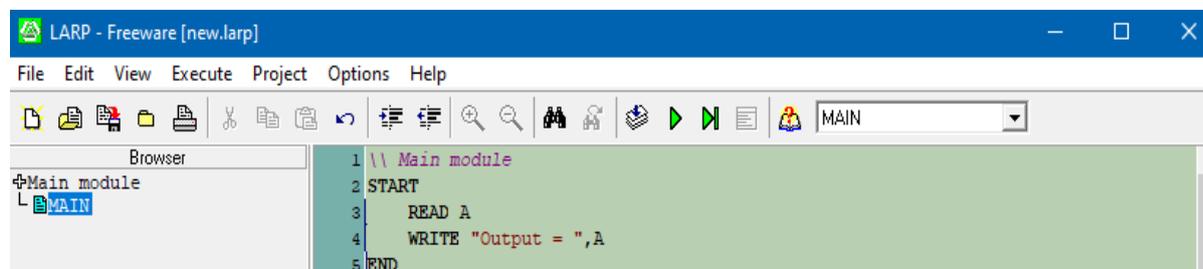


Fig. 13 (Pseudocode in LARP)

- Enter the input value on the console
- Press **"Enter"** key. Output will be displayed
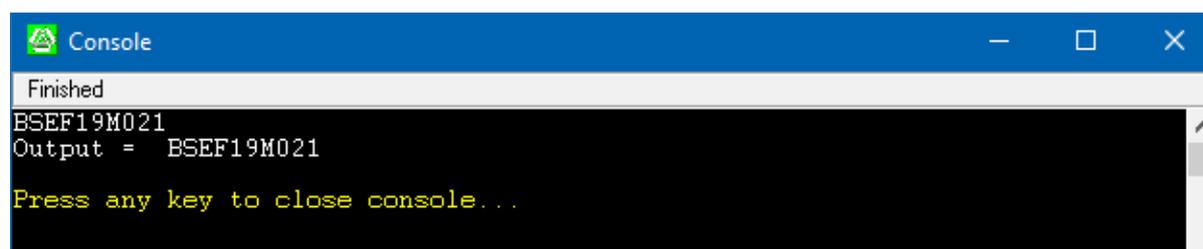


Fig. 14 (LARP Console)

**Reading Multiple Inputs:**

- On the text editor in LARP, type **"READ {variables}"** OR drag the **"READ {variable_list}"** template from the **"Templates"** column
- Replace the variable list with your desired comma separated variable names
- On the next line type **"WRITE"** following with comma separated variable names
- On the ribbon, click the execute button OR press **"F7"** key
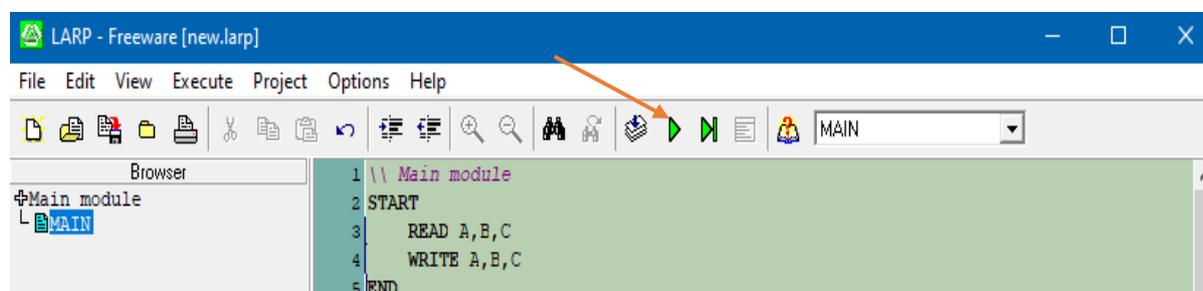


Fig. 15 (Pseudocode in LARP)

- Enter space separated input values on the console
- Press **"Enter"** key. Output will be displayed
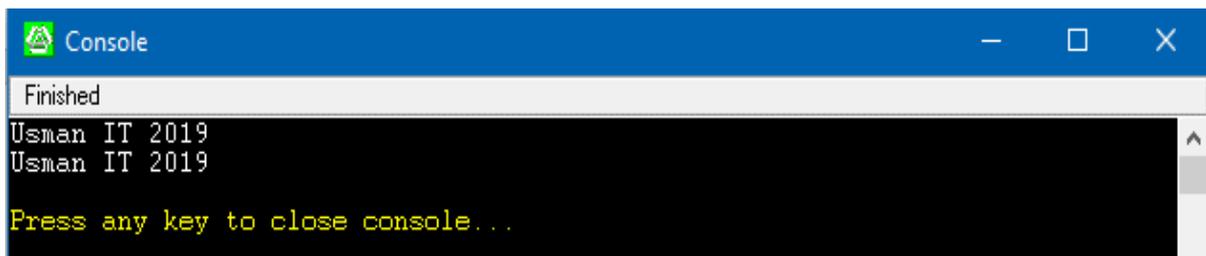
**IF ELSE Statement:**

The IF/ELSE statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.

- Drag **"IF {condition} THEN {instructions} ELSE IF {condition} THEN {instructions} ENDIF"** on text editor from **"Templates"** column
- Write a code which read a number from user and then prints the number in text form as shown in figure
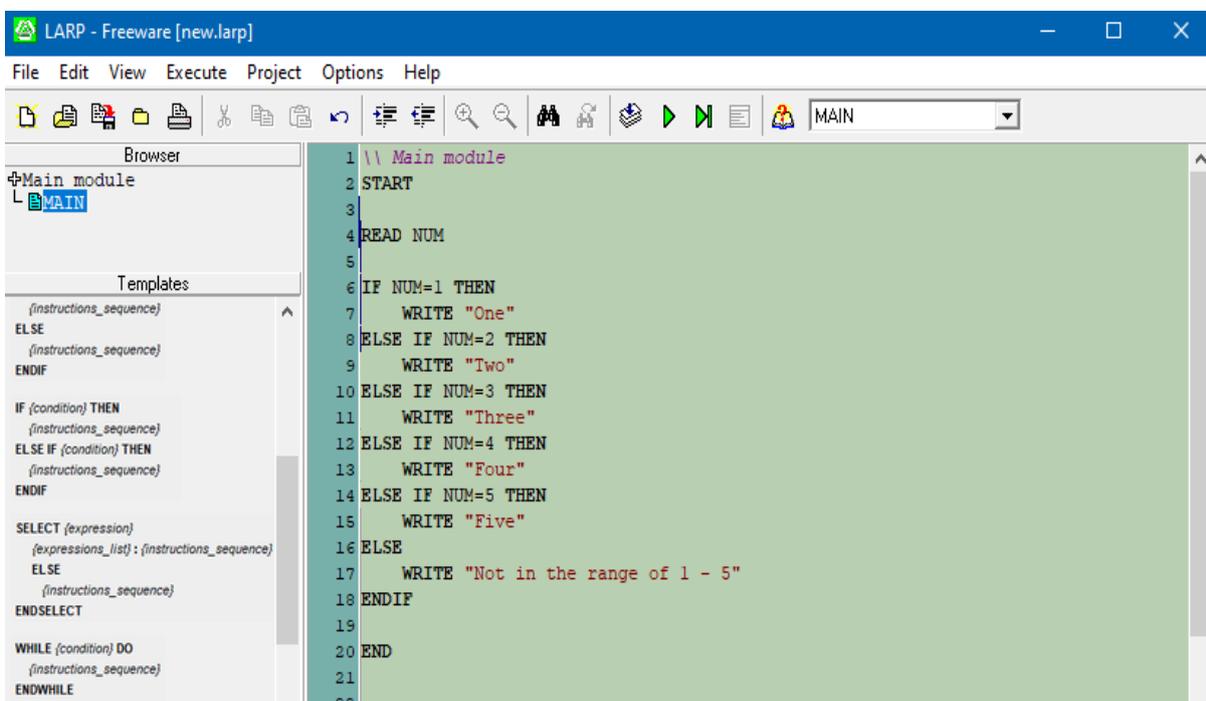


Fig. 17 (Pseudocode in LARP)

- Enter input value on the console
- Press **"Enter"** key. Output will be displayed
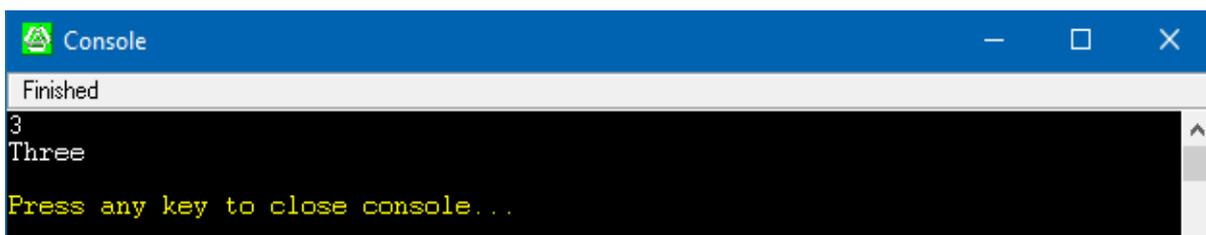


Fig. 18(LARP Console)

**Loop Statements:**

A Loop executes series of statements repeatedly for a specified number of times or until specified conditions are met.

- Drag **"WHILE {condition} DO {instructions} ENDWHILE"** on the text editor from the **"Templates"** box
- Replace the condition & instruction so that console prints numbers from 1 to 5
- Similarly drag **"FOR {variable} = {initial value} TO {final value} DO {instructions} ENDFOR"**
- Replace the values & instruction so that console prints numbers from 6 to 10
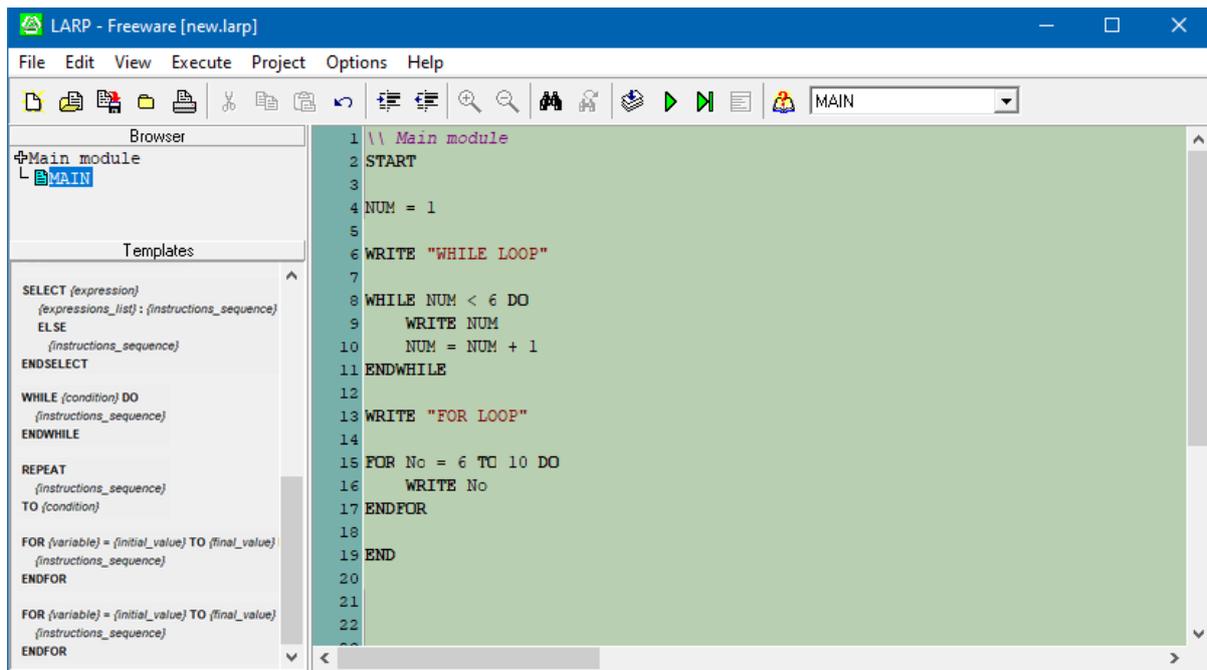


Fig. 19(Pseudocode in LARP)

- Press **"F7"** key. Output will be displayed on the console
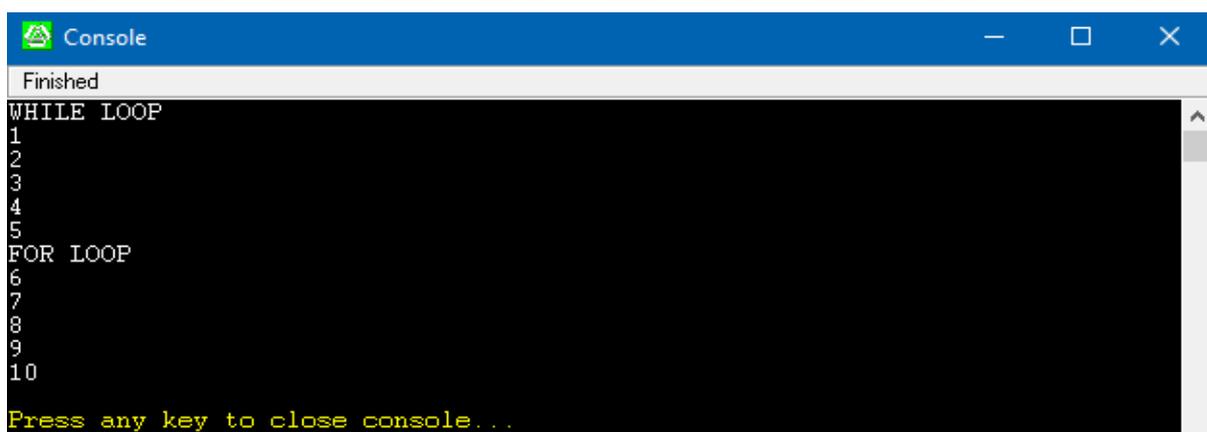


Fig. 20 (LARP Console)

**Flowchart**

| Symbol | Name | Function |
|---|---|---|
|  | Start/End | Oval represents a Starting or Ending point |
|  | Process | A rectangle represents a process |
|  | Input/Output | A parallelogram represents an input or output |
|  | Decision | A diamond represents a decision |
|  | Arrow | Arrow shows a relationship between representative shapes |
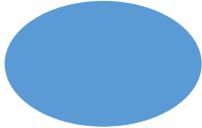
**Input / Output:**

- Open flowchart project in LARP
- Drag **"Parallelogram"** from the **"Templates"** box and drop it on the connector (where arrow is pointing in the figure)
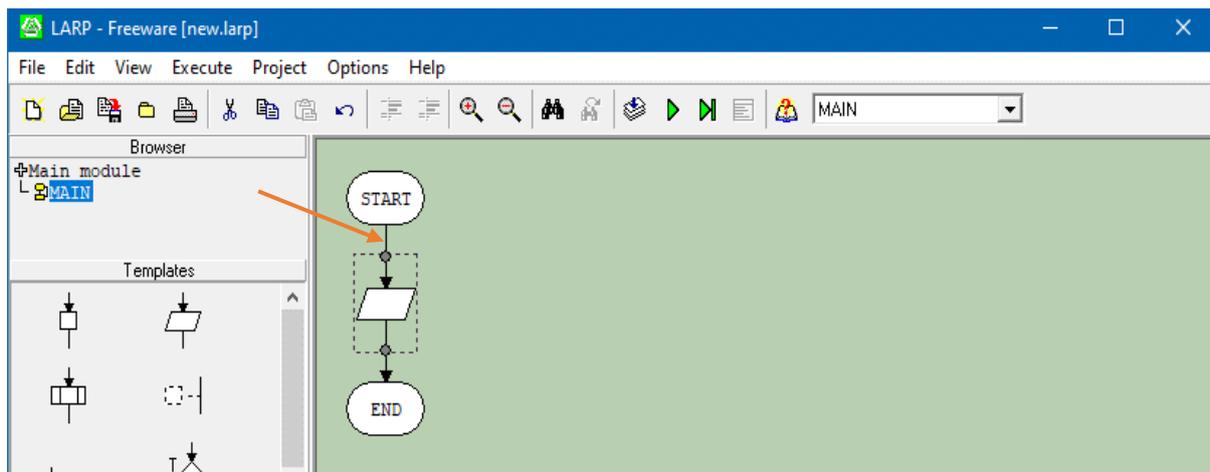


Fig. 21 (LARP Flowchart)

- Double click the parallelogram
- Edit dialog will open. Type **"Num"** under **"Read"** column
- Click **"OK"**
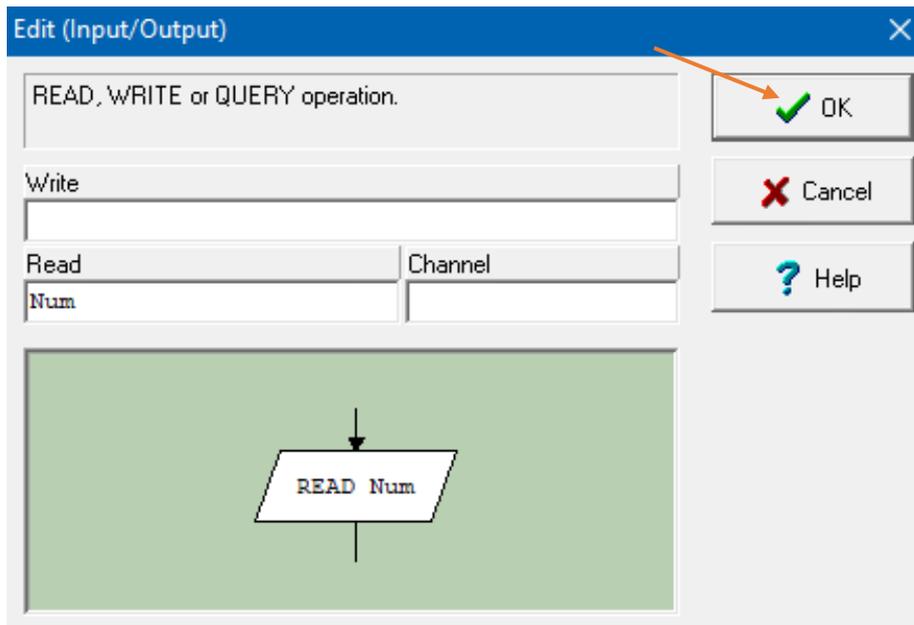
Fig. 22 (LARP Flowchart)

- Drag & drop another parallelogram
- Double click on it
- Type **"Output =, Num"** under **"Write"** column
- Click **"OK"**
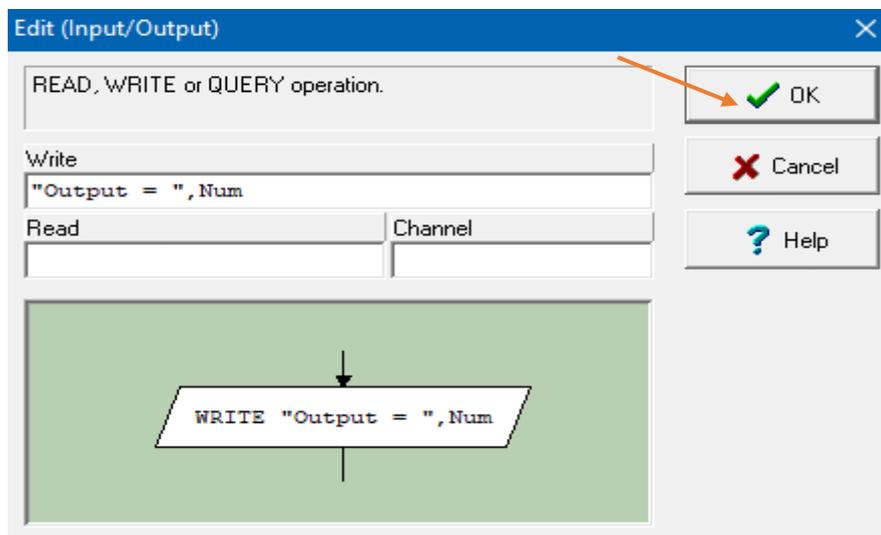


Fig. 23 (LARP Flowchart)

- On the ribbon, click the execute button OR press **"F7"** key
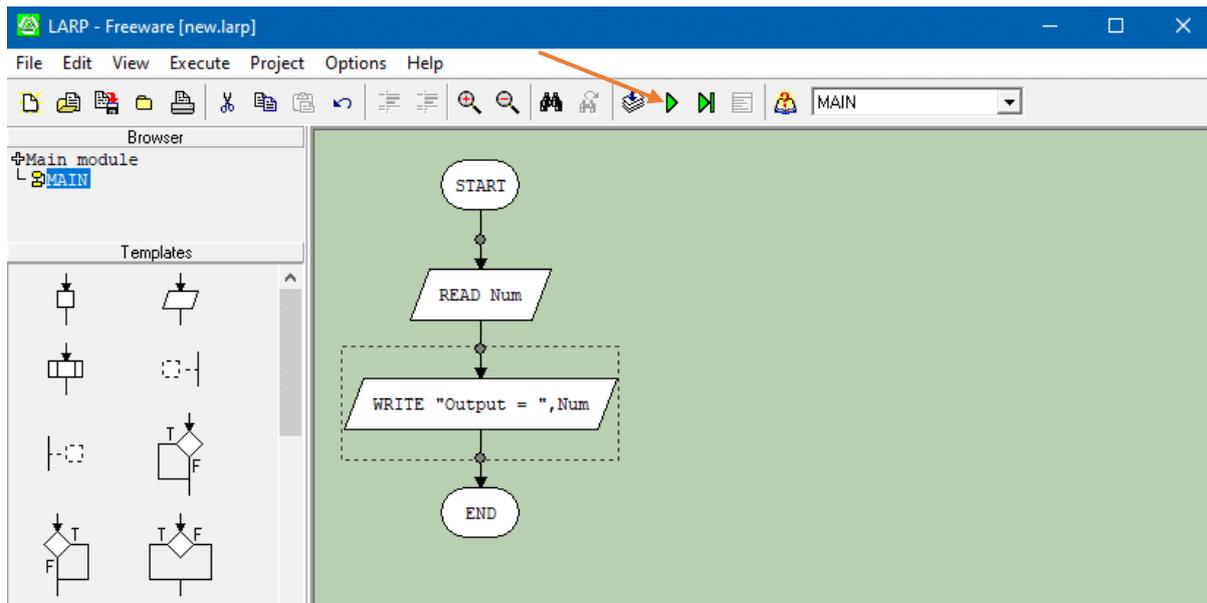
- Enter the input value on console
- Press **"Enter"** key
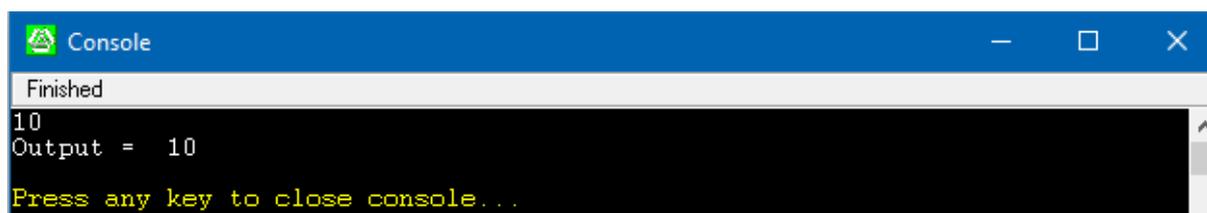
**Decision**

Decision symbol is used to make decisions. It can be used for IF ELSE conditional, WHILE Loop, FOR loop etc. In LARP different decision templates are provided under **"Templates"** box.

**IF ELSE Conditional:**

- Read a number from user with variable name **"Num"** using Input/Output symbol
- Drag IF ELSE Conditional from the **"Templates"** box
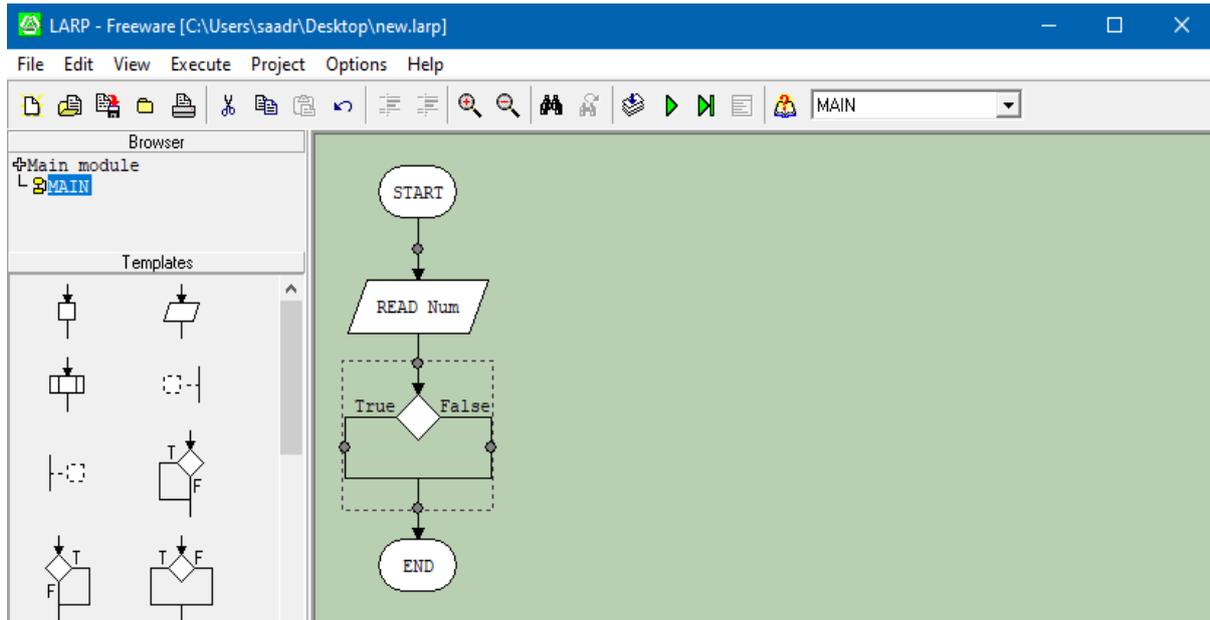
Fig. 26 (LARP Flowchart)

- Double click it. Edit dialog will open
- Type **"Num > 5"** under **"Condition"** column
- Click **"OK"**
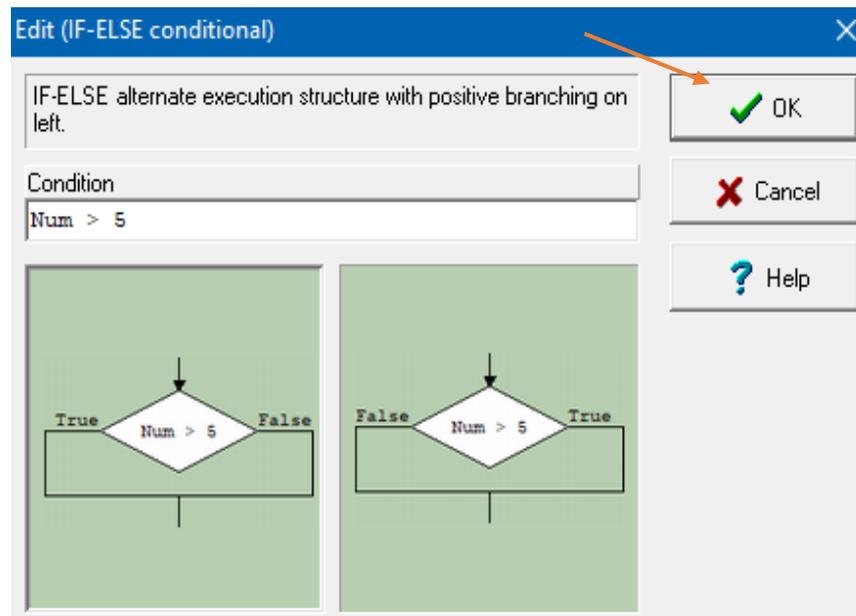


Fig. 27 (LARP Flowchart)

- Display **"Greater than 5"** if the condition is true else display **"Less than or equal to 5"**
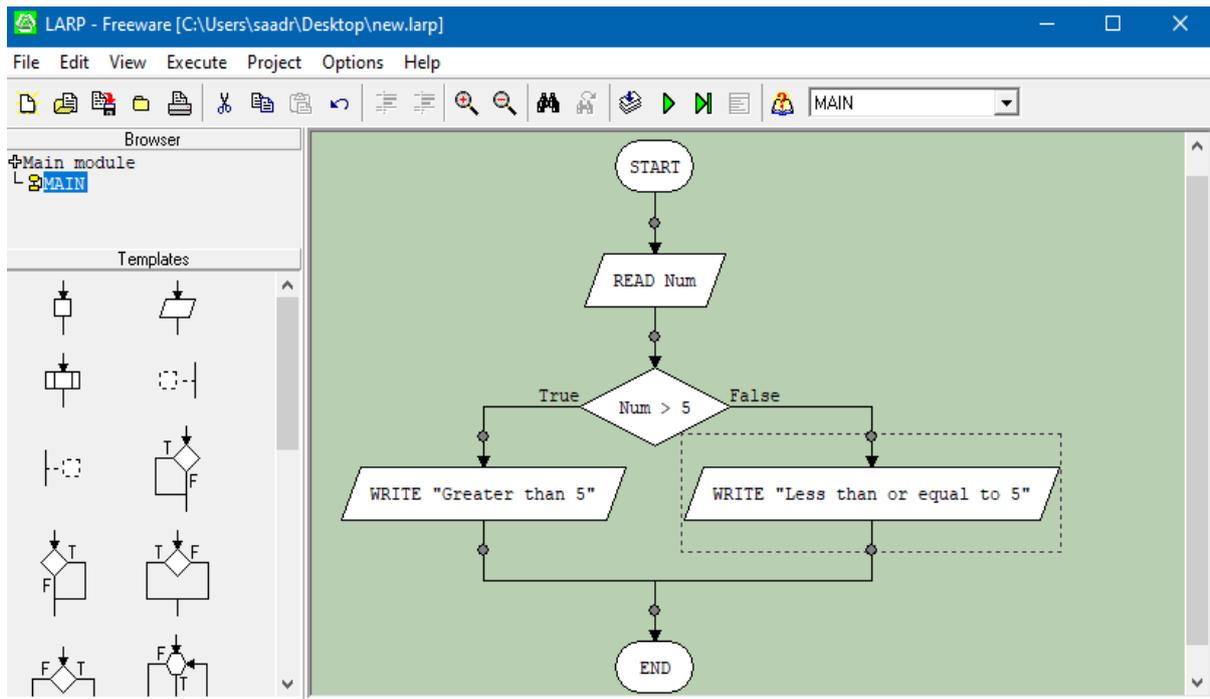
Fig. 28 (LARP Flowchart)

- Press **"F7"** key
- Enter the input value on console
- Press **"Enter"** key
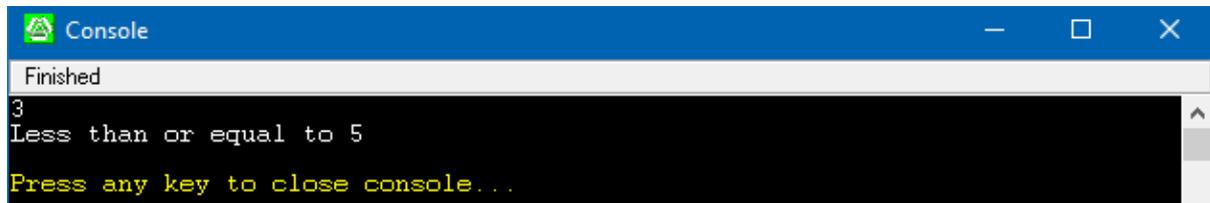


Fig. 29 (LARP Console)

**FOR Loop:**

A FOR Loop is a statement for specifying iteration, which allows the code to be executed repeatedly.

- Drag FOR Loop template from the **"Templates"** box
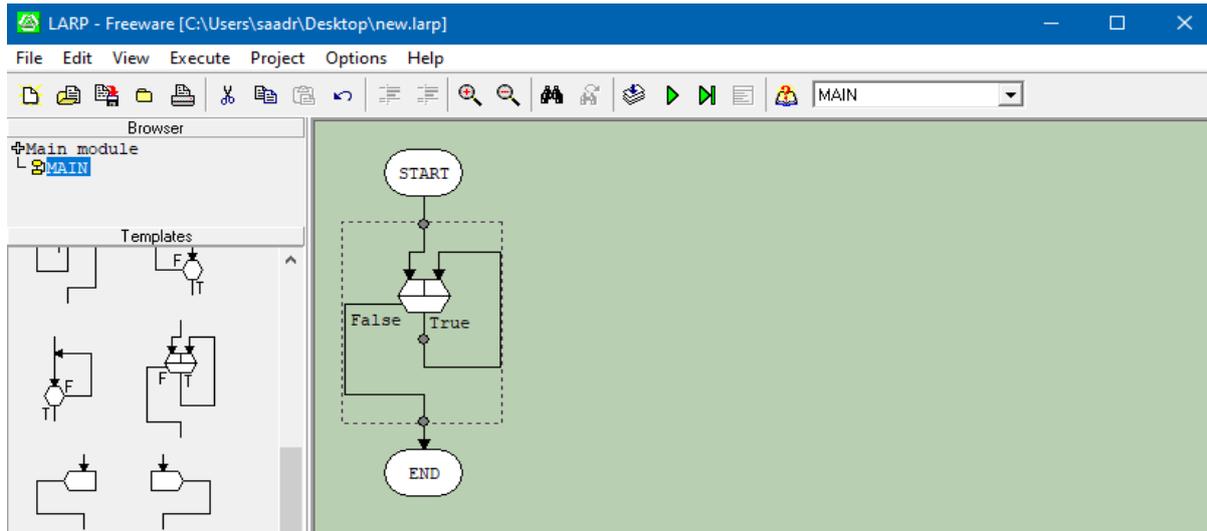
Fig. 30 (LARP Flowchart)

- Double click it. Edit dialog will open
- Type **"Num"** under **"Variable"** column
- Type **"2"** under **"Increment"** column
- Type **"1"** under **"Initial value"** column
- Type **"10"** under **"Final value"** column
- Click **"OK"**
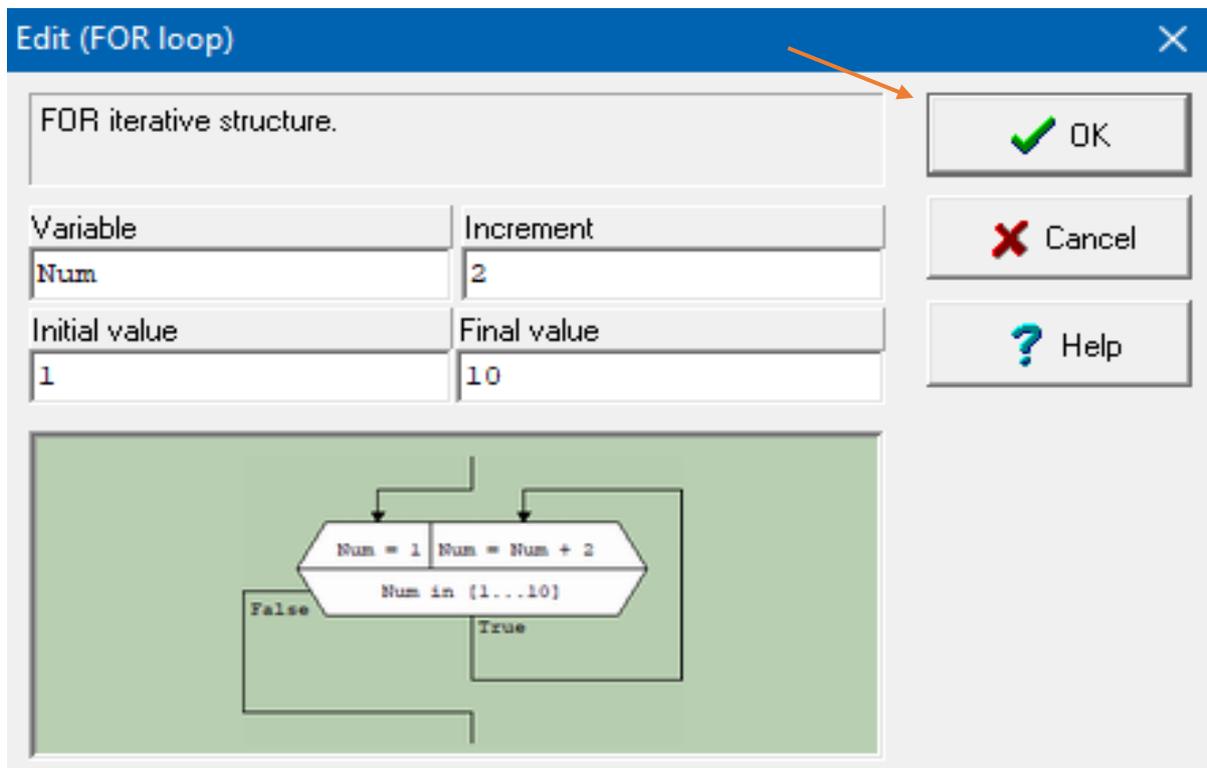


Fig. 31 (LARP Flowchart)

- Display **"Num"** if the condition is true. It will display odd numbers up to 10
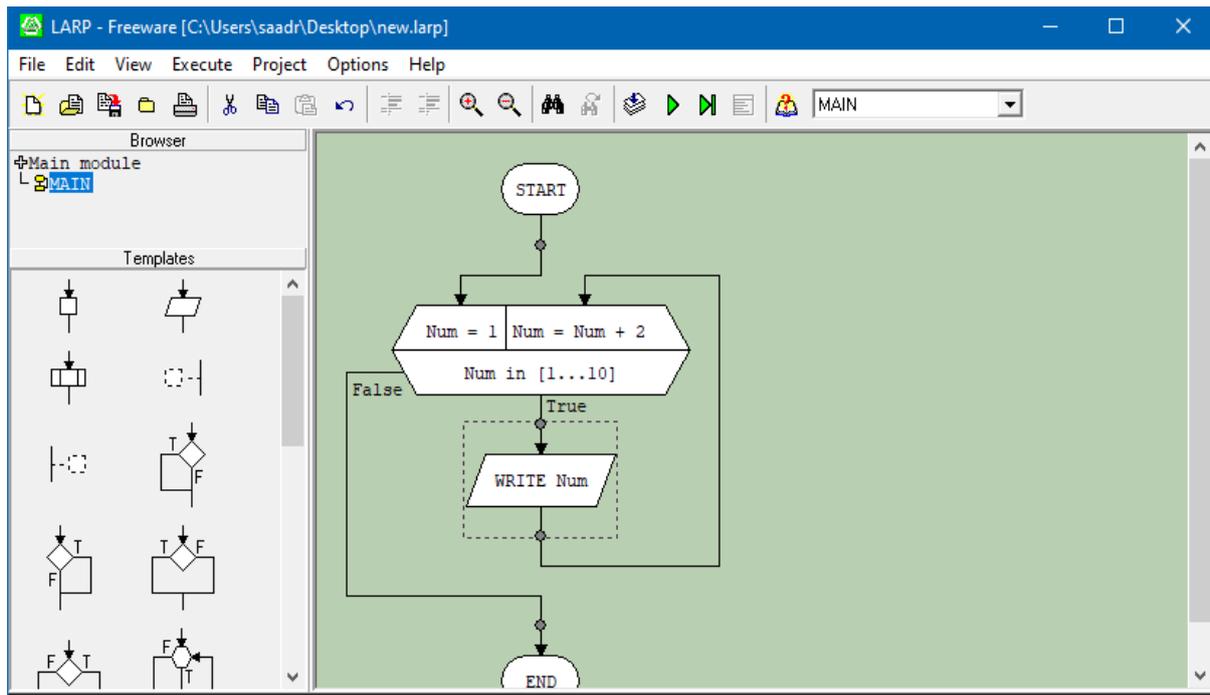
Fig. 32 (LARP Flowchart)

- Press **"F7"** key. Output will be displayed
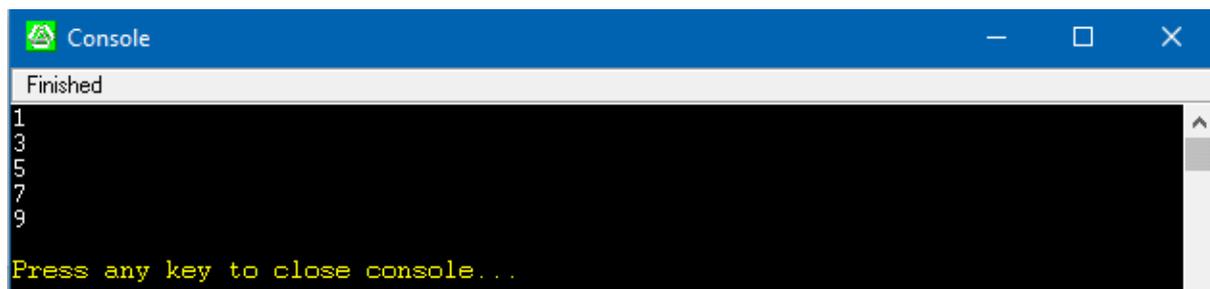


Fig. 33 (LARP Console)

**Saving a file in LARP:**

- On the ribbon, click Save button OR Press **"CTRL + S"**
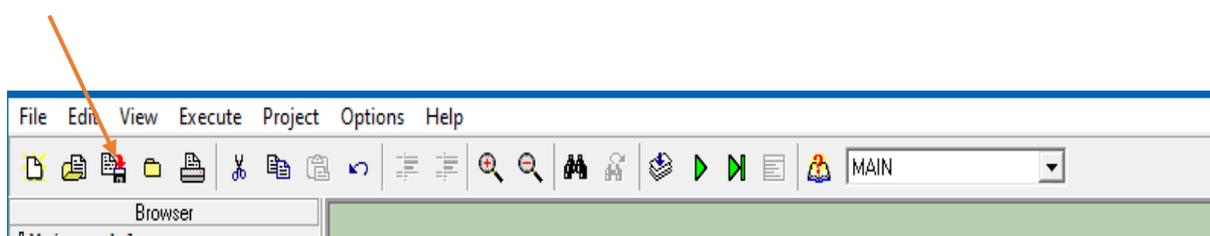


Fig. 34 (Save LARP File)

- Save dialog will open. Choose your desired path
- Click **"Save"**
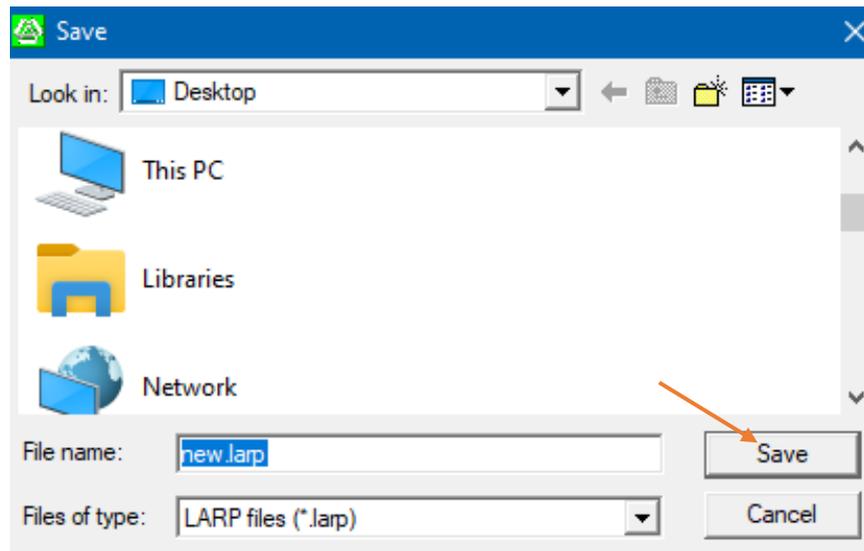
Fig. 35 (Save LARP File)

**C program in Microsoft ® Visual Studio 2022:**

- Open Microsoft ® Visual Studio 2022
- Click on **"Create a new project"**
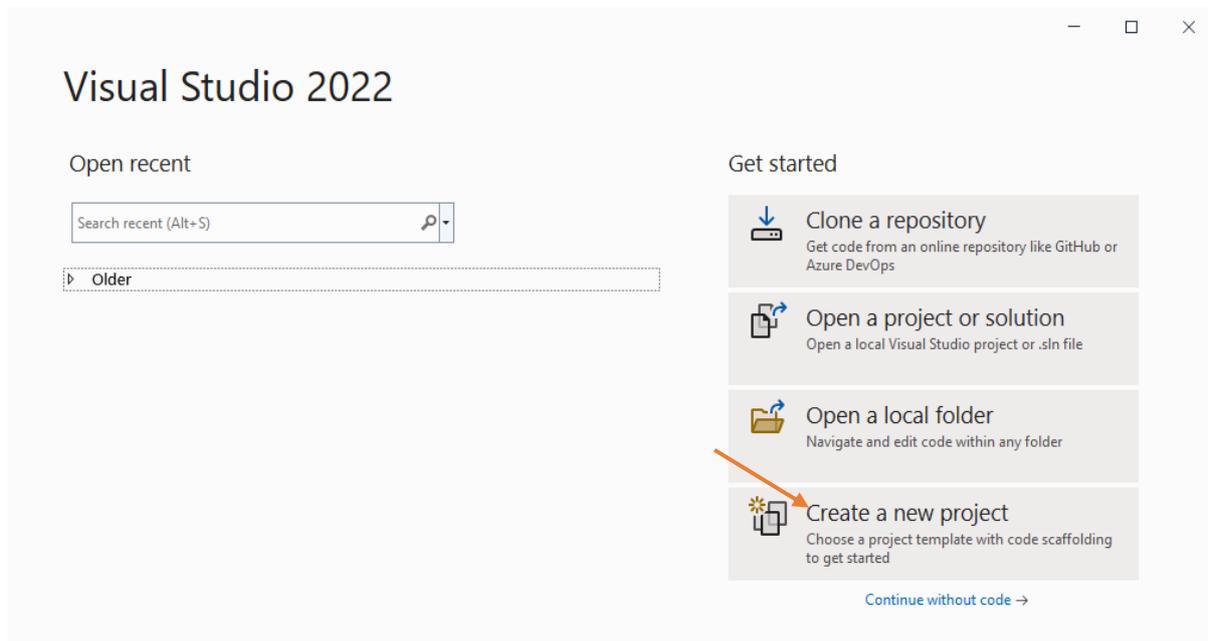


Fig. 36 (Microsoft ® Visual Studio 2022)

- Select the **"Empty Project"**
- Click **"Next"**
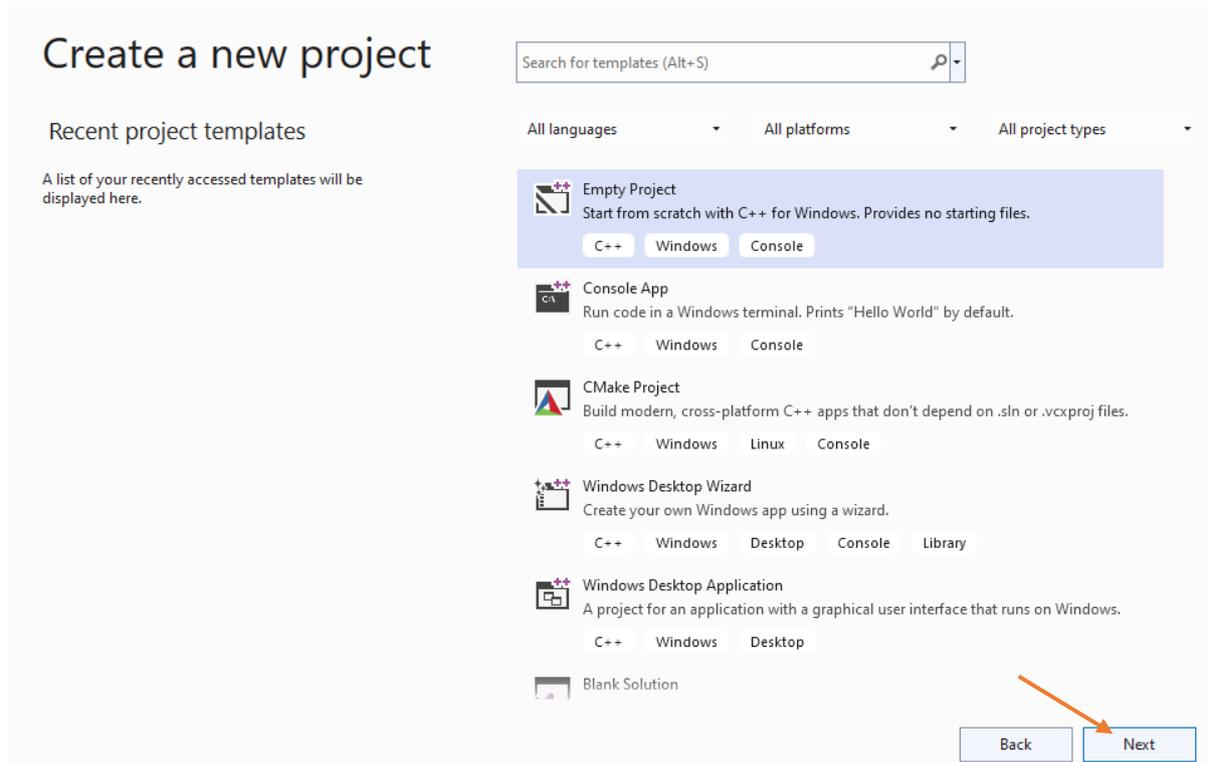
Fig. 37 (Microsoft ® Visual Studio 2022)

- Enter **"First Program"** in Project Name
- Click **"Create"**
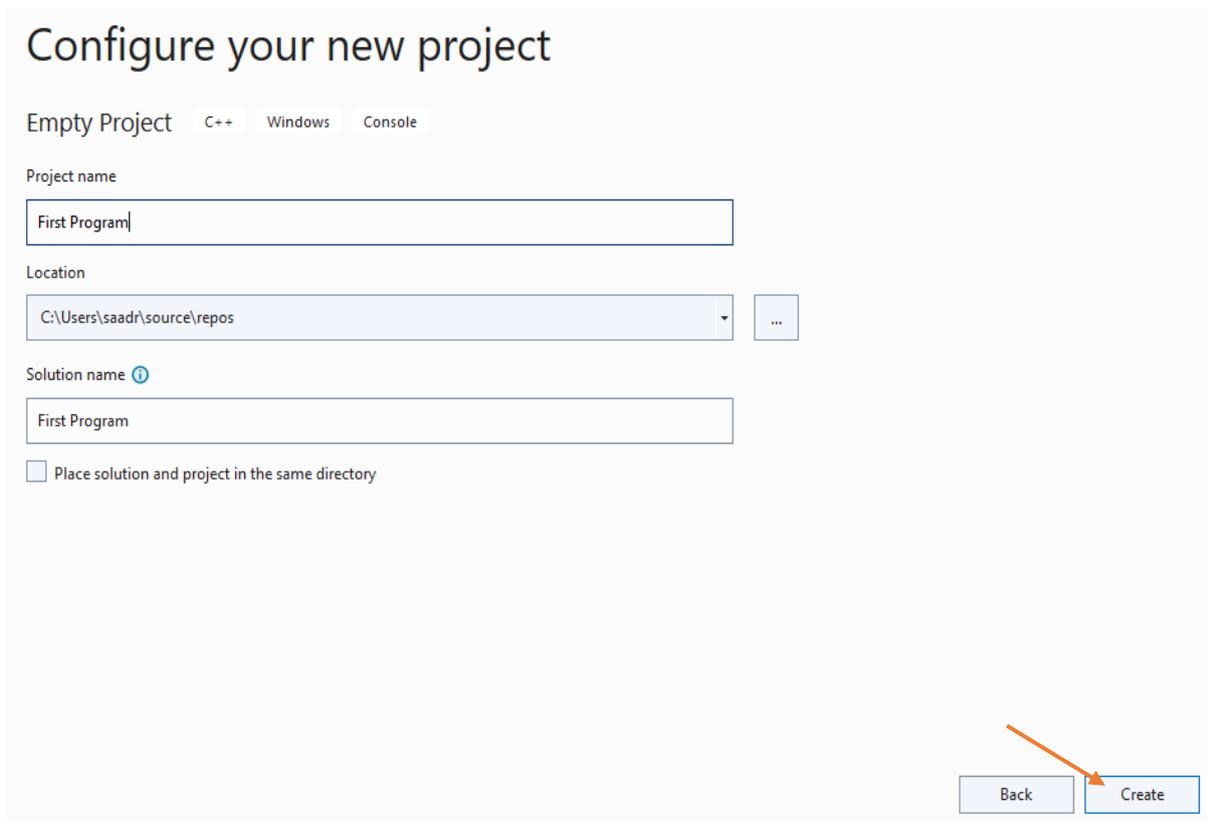


Fig. 38 (Microsoft ® Visual Studio 2022)

- On right side, Under **"Solution Explorer"**
- Right click **"Source Files"**
- Select **"Add"**
- Click **"New Item"**
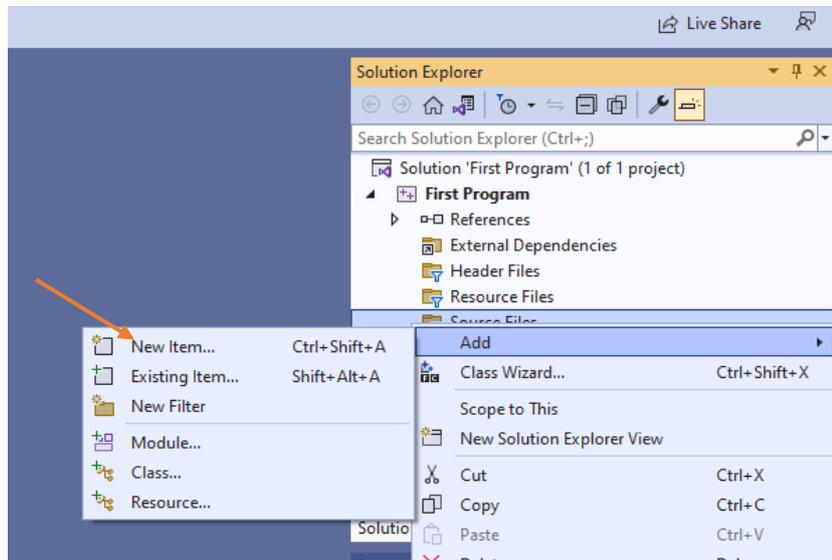


Fig. 39 (Microsoft ® Visual Studio 2022)

- Select **"C++ File (.cpp)"**
- Name it **"Source.c"**
- Click **"Add"**
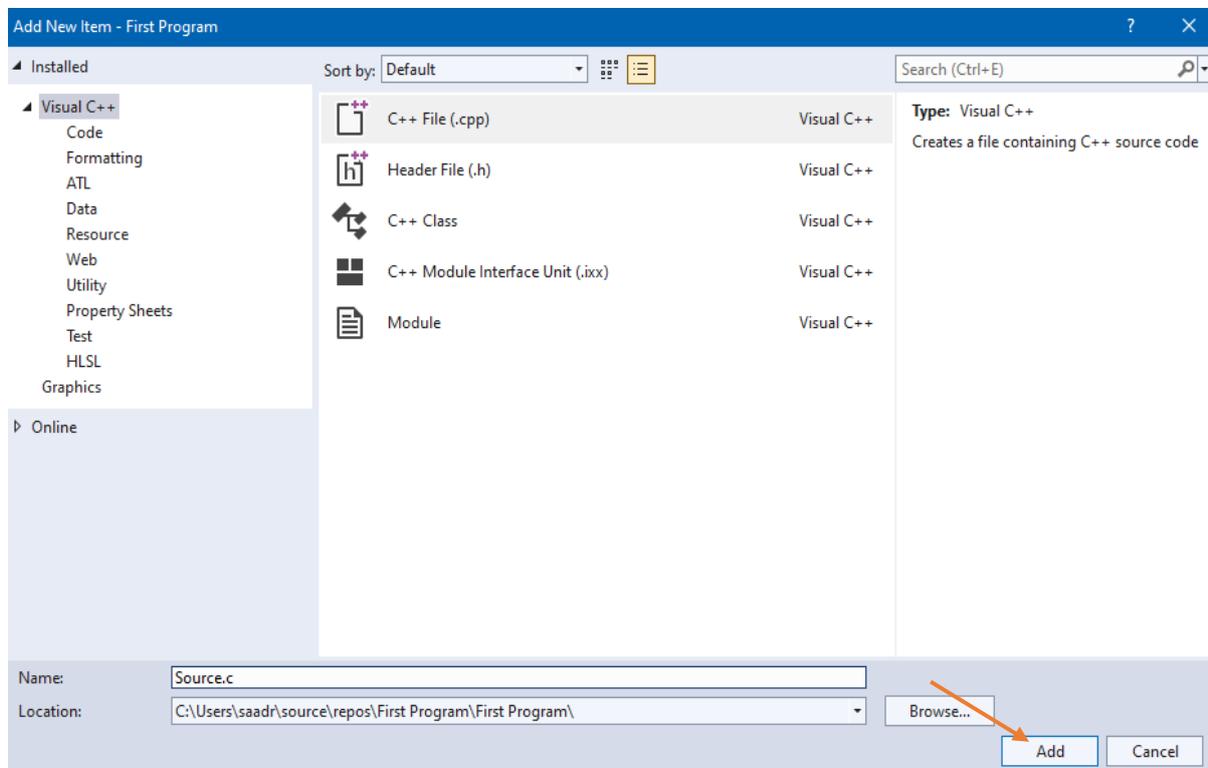


Fig. 40 (Microsoft ® Visual Studio 2022)

- Write a C program to display **"Hello World"**
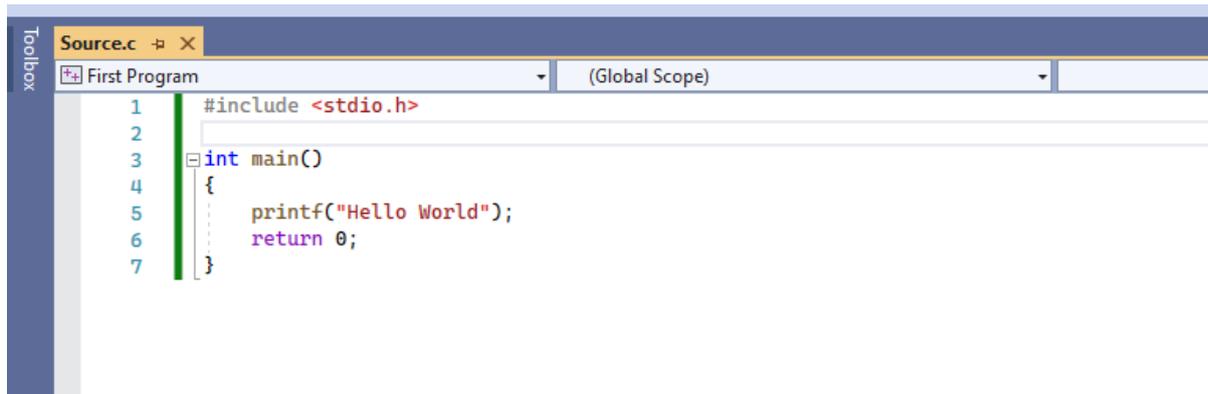


Fig. 41 (Microsoft ® Visual Studio 2022)

- On the menu bar select **"Debug"**
- Click **"Start Without Debugging"** OR Press **"CTRL + F5"**
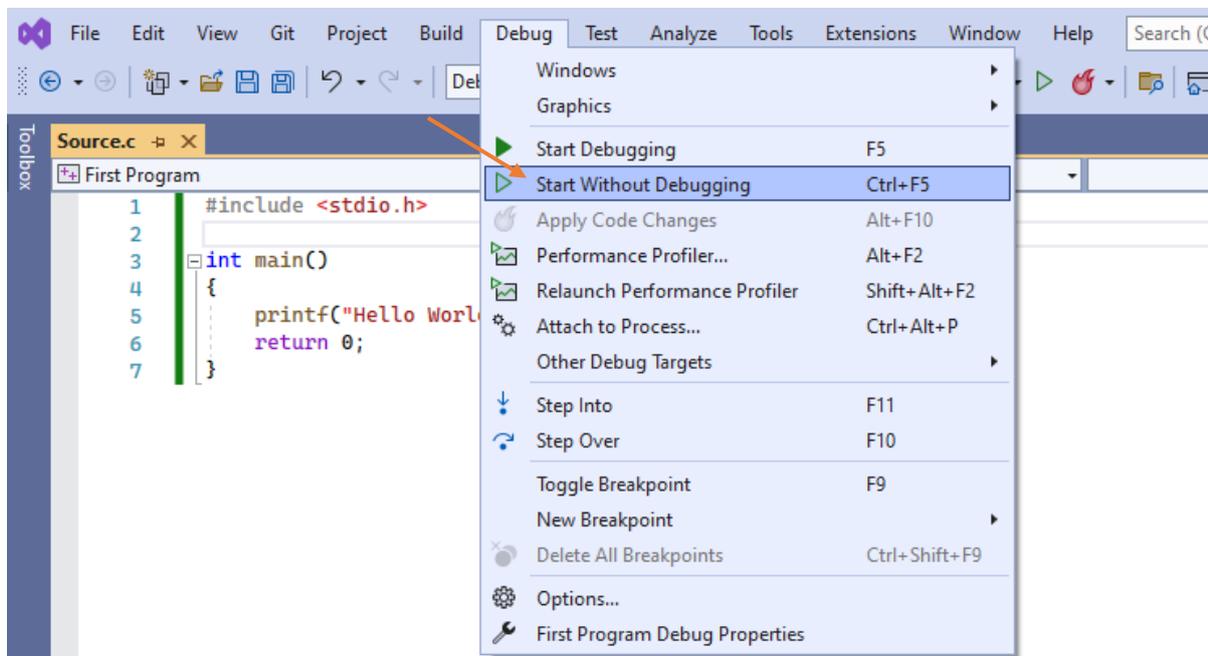


Fig. 42 (Microsoft ® Visual Studio 2022)

Output will be displayed on the console.
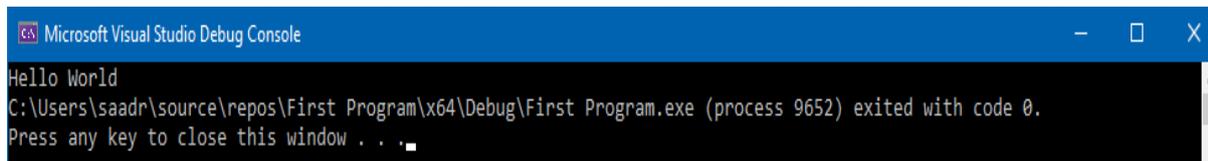


Fig. 43 (Microsoft ® Visual Studio Console)

**Operators in C:**

| Operator Type | Name | Operator |
|---|---|---|
| Unary Operators | Increment | ++ |
| | Decrement | -- |
| Arithmetic Operators | Addition | + |
| | Subtraction | - |
| | Multiplication | * |
| | Division | / |
| | Modulus | % |
| Logical Operators | AND | && |
| | OR | \|\| |
| | NOT | ! |
| Relational Operators | Less than | < |
| | Greater than | > |
| | Less than Equal to | <= |
| | Greater than Equal to | >= |
| | Equal to | == |
| | Not Equal to | != |
| Bitwise Operators | Bitwise AND | & |
| | Bitwise OR | \| |
| | Bitwise XOR | ^ |
| | Bitwise Complement | ~ |
| | Shift left | << |
| | Shift right | >> |
| Ternary Operators | Conditional | ?: |

**Operator Precedence in C:**

Operator precedence determines the grouping of terms in an expression and decides how an expression is evaluated. Operators with highest precedence appear at the top of the table, those with lowest appear at the bottom.

| Operator Type | Operators |
|---|---|
| Postfix | (), [], ->, ++, -- |
| Unary | +, -, !, ~, ++, -- |
| Multiplicative | *, /, % |
| Additive | +, - |
| Shift | <<, >> |
| Relational | <, <=, >, >= |
| Equality | ==, != |
| Bitwise AND | & |
| Bitwise XOR | ^ |
| Bitwise OR | \| |
| Logical AND | && |
| Logical OR | \|\| |
| Conditional | ?: |

**Integer Division:**

Division in which the fractional part is discarded is called Integer Division. For example

$11/2 = 5 + ½$. In Integer Division, $11/2 = 5$. Demonstration in C programming is shown below.

- Open Web browser
- In search bar type, **"Online C compiler"**
- Open first site



Fig. 44 (Online Compiler)

**Laboratory 01 – Introduction to Programming, Algorithms and C**          Page **26** of **31**
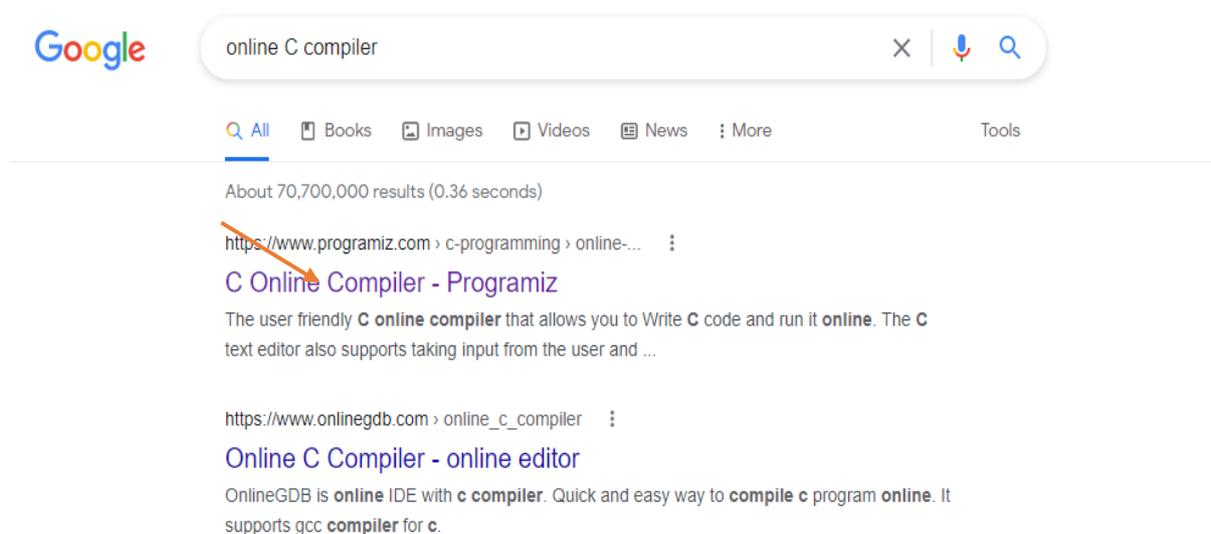
- Write a simple program which divides an odd integer by 2 as shown in figure
- Click the **"Run"** button
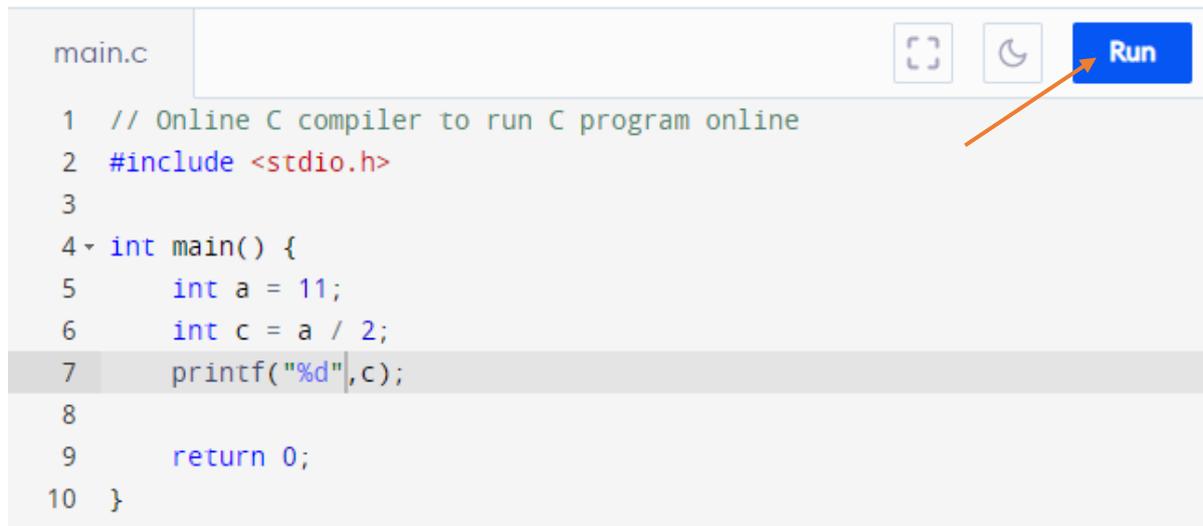


Fig. 45 (Online Compiler)

Output will be shown next to the editor screen.
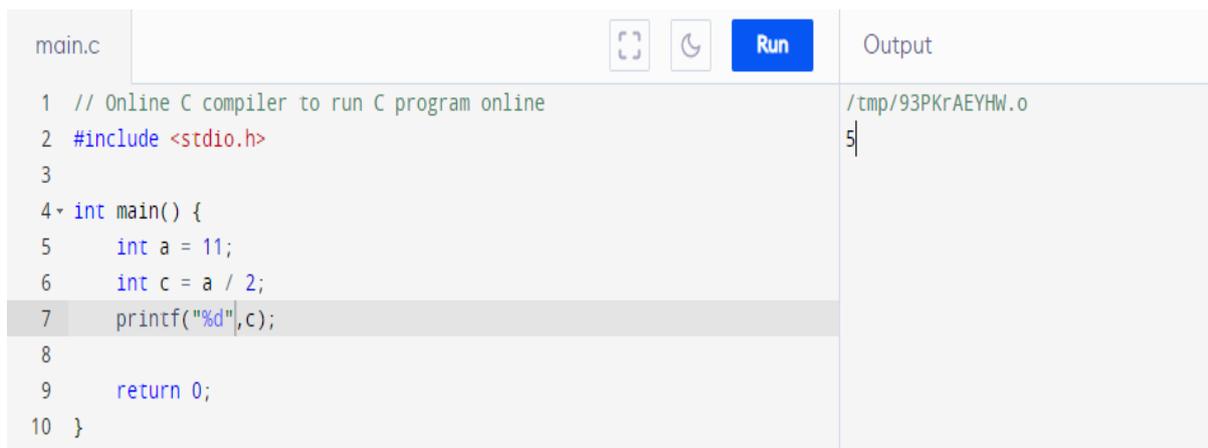


Fig. 46 (Online Compiler)

**Task 01: Pseudocode**                                    **[35 minutes / 30 marks]**

**Part (a):**                                              **[15 minutes / 10 marks]**

Write a Pseudocode in LARP software which:

- Reads a number as an input from user
- Check if the number is positive or negative
- Displays **"Number is positive"** in case of positive number and **"Number is negative"** in case of negative number

**Part (b):**                                             **[20 minutes / 20 marks]**

Write a Pseudocode in LARP software which:

- Uses **"FOR Loop"** OR **"WHILE Loop"** template
- Uses **"IF ELSE_IF"** template
- Displays the following pattern by using above templates:
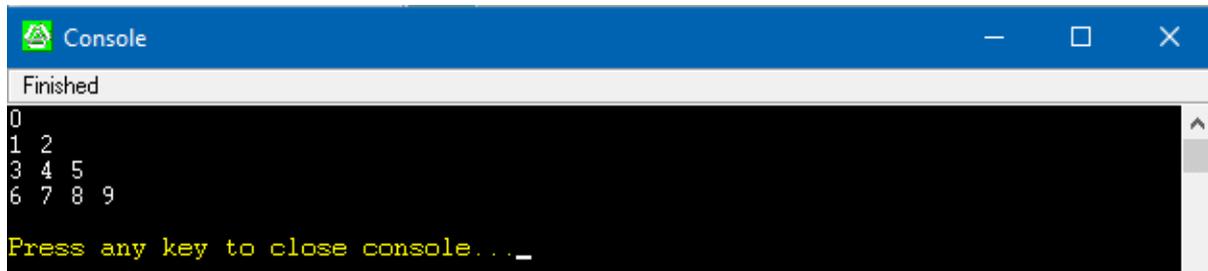


Fig. 47 (In-Lab Task)

**Task 02: Flowchart.**                                   **[15 minutes / 15 marks]**

Create a Flowchart in LARP software which:

- Read two numbers from a user
- Display the numbers on console
- Check which number is maximum
- Display the maximum on console.

Output should be in the following format:
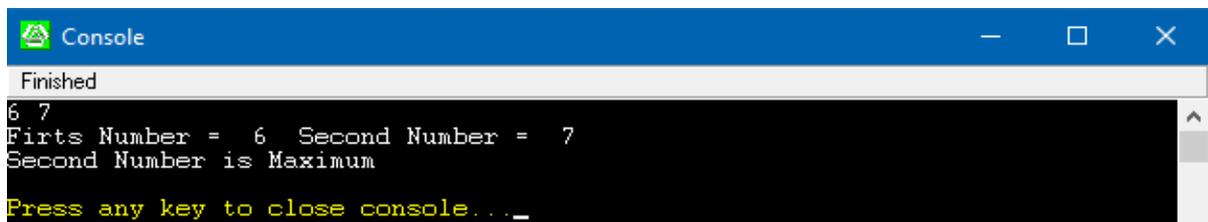


Fig. 48 (In-Lab Task)

**Task 03: Visual Studio.**                               **[10 minutes / 5 marks]**

Create a C program in Visual Studio which displays your **"Name"** and **"Roll No"** on the Console.

**Post-Lab Activities:**

**Task 01: Flowchart**                                   **[Estimated 30 minutes / 30 marks]**

Create a Flowchart in LARP software which:

- Finds **"10"** Prime numbers next to your Roll No

e.g., if your Roll No is **"6"** then 10 prime numbers which your program finds will be **"7, 11, 13, 17..."**

- Sum the Prime Numbers
- Displays the **"Sum"** on the console
- Submit **".larp"** file named your **"Roll No"** on Google Classroom

## Submissions:

- For In-Lab Activity: Save the files on your PC. TA's will evaluate the tasks offline.
- For Post-Lab Activity: Submit the .larp file on Google Classroom and name it to your roll no.

## Evaluations Metric:

- All the lab tasks will be evaluated offline by TA's
- **Division of In-Lab marks:**                                        **[50 marks]**
  - Task 01 Part (a): Pseudocode                                [10 marks]
  - Task 01 Part (a): Pseudocode                                [20 marks]
  - Task 02: Flowchart                                          [15 marks]
  - Task03: Visual Studio                                       [05 marks]
- **Division of Post-Lab marks:**                                      **[30 marks]**
  - Task01: Flowchart                                           [30 marks]

## References and Additional Material:

- Microsoft ® Visual Studio 2022 Installation
  https://visualstudio.microsoft.com/

- LARP 3.0 Installation
  https://en.freedownloadmanager.org/Windows-PC/LARP.html

- Microsoft ® Visual Studio Documentation
  https://docs.microsoft.com/en-us/visualstudio/windows/?view=vs-2022

- LARP Documentation
  http://larp.marcolavoie.ca/en/documentation/documentation.htm

## Lab Time Activity Simulation Log:

- Slot – 01 – 00:00 – 00:15:      Class Settlement
- Slot – 02 – 00:15 – 00:30:      Pseudocode
- Slot – 03 – 00:30 – 00:45:      Pseudocode
- Slot – 04 – 00:45 – 01:00:      Flowchart
- Slot – 05 – 01:00 – 01:15:      Flowchart
- Slot – 06 – 01:15 – 01:30:      C Programming Environment
- Slot – 07 – 01:30 – 01:45:      Operators & Integer Division
- Slot – 08 – 01:45 – 02:00:      In-Lab Task
- Slot – 09 – 02:00 – 02:15:      In-Lab Task
- Slot – 10 – 02:15 – 02:30:      In-Lab Task
- Slot – 11 – 02:30 – 02:45:      In-Lab Task
- Slot – 12 – 02:45 – 03:00:      In-Lab Tasks Assessment