

Sudoku Solver

Aim:

To solve the 9X9 Sudoku Grid.

Input:

A Partially filled 9 X 9, 2D Array.

Output:

A Completely solved Sudoku.

Edge Cases:

1. If the grid is completely pre-filled/solved already,
Program will exit with appropriate Message.
2. If the Values are Incorrect and a Sudoku grid cannot be made then,
Program will exit with Appropriate Message.

Technologies Used:

Open CV, Visual Studio 2019, C++

How to run the Program:

Installation of Required Software's:

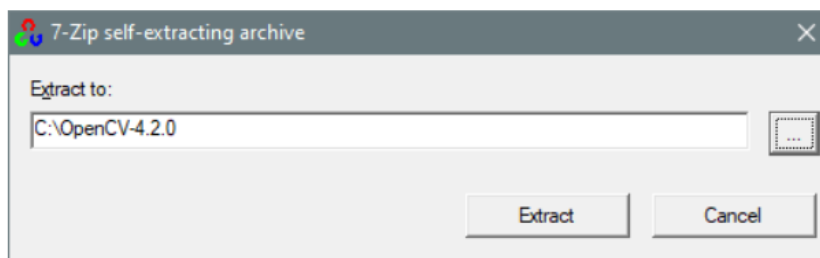
You can Refer to my YouTube Channel: <https://youtu.be/riArxlmRMoQ>

Step 1: Download and Install Open CV from official Website

Or Git Repository: <https://github.com/opencv/opencv/releases/tag/4.1.2>

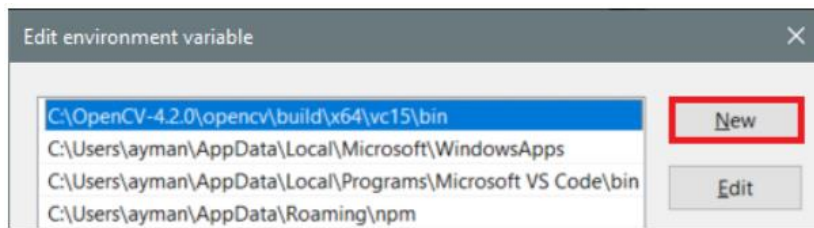
Download the .exe file

Extract it



Step 2: Add Open CV libraries to your System Path

Once OpenCV is correctly installed in your folder, you now have to add the binaries `C:\OpenCV-4.2.0\opencv\build\x64\vc15\bin` to your system path, so you can have access to OpenCV executables easily through your command line.



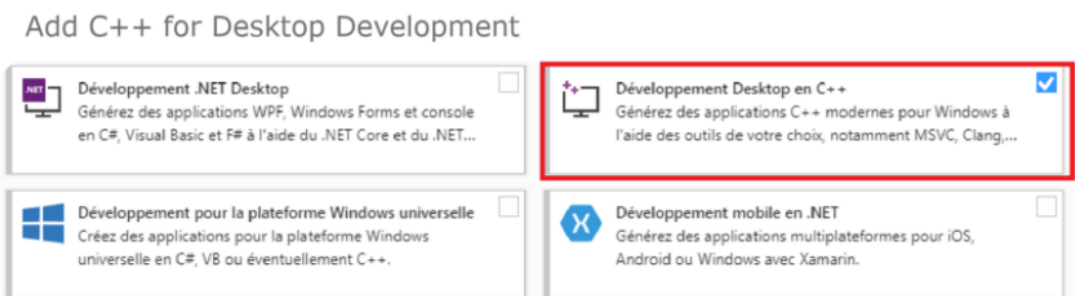
Editing the system path

Step 3: Downloading and Installing Visual Studio 2019

Download Community Version from official Website

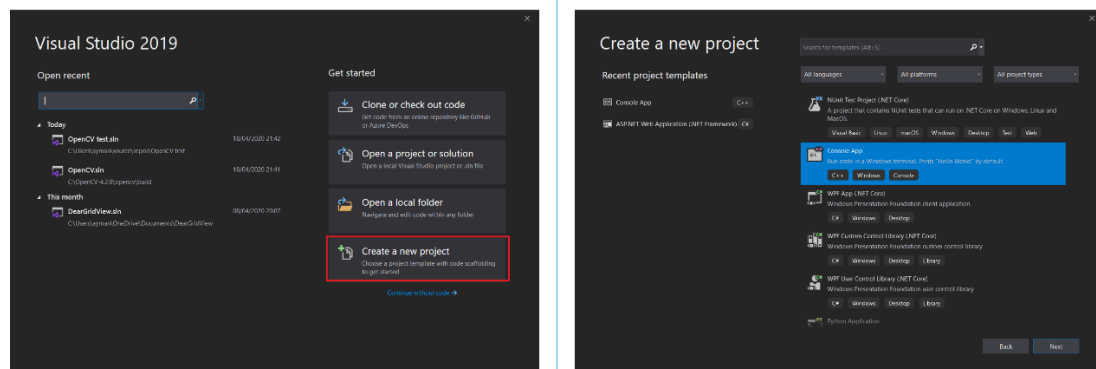
Link: <https://visualstudio.microsoft.com/>

Check the box to install “Development Desktop on C++”



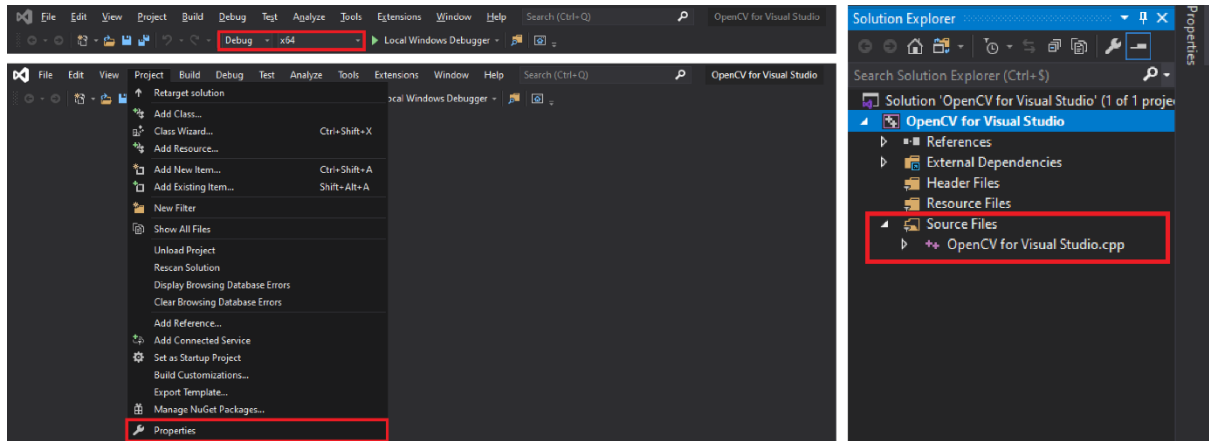
Step 4: Configure a Visual Studio project to run OpenCV

Open Visual Studio 2019, choose to create a new project and go for the C++ Console App template.



Once the project created you should have a layout with a solution explorer to the right having one source file and, in the window, upper ribbon you should see Debug for x86 platforms, meaning that the project will build and run-in debug mode targeting x86 windows architectures.

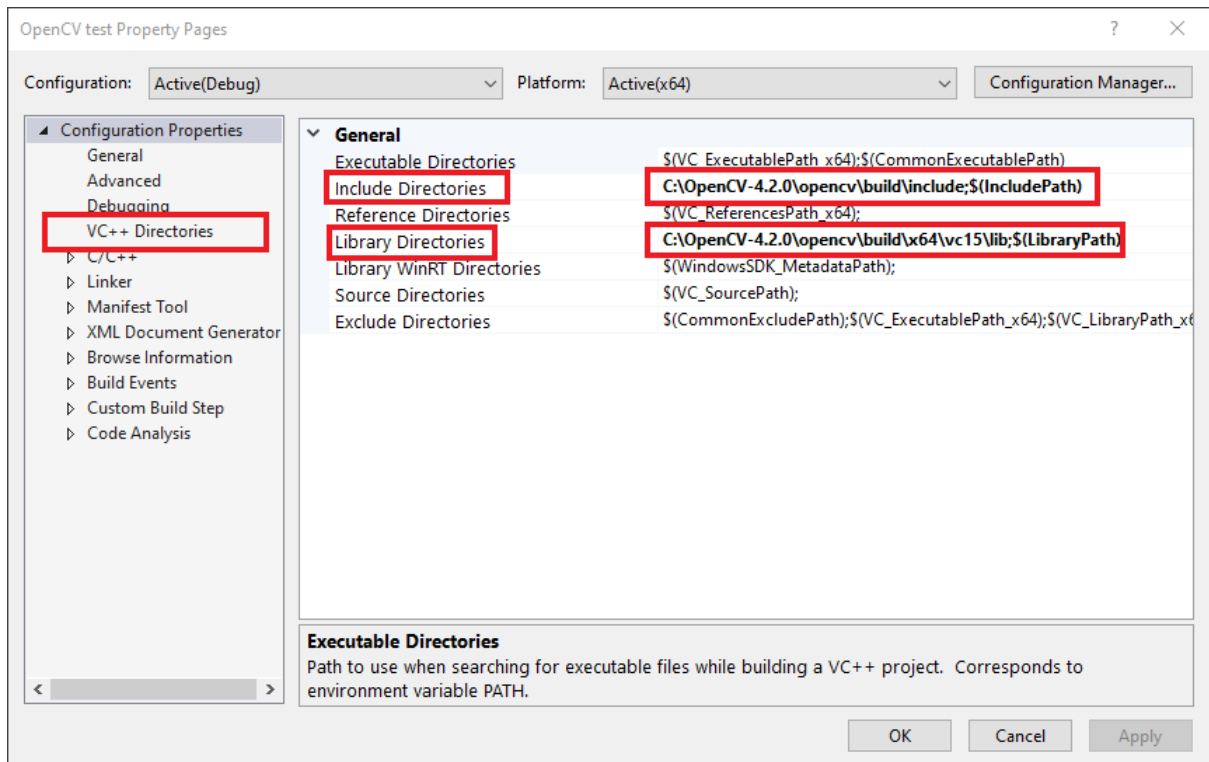
- First, you should change the solution platforms from x86 to x64
- Secondly, you have to change the **Project Properties** to add the OpenCV libraries



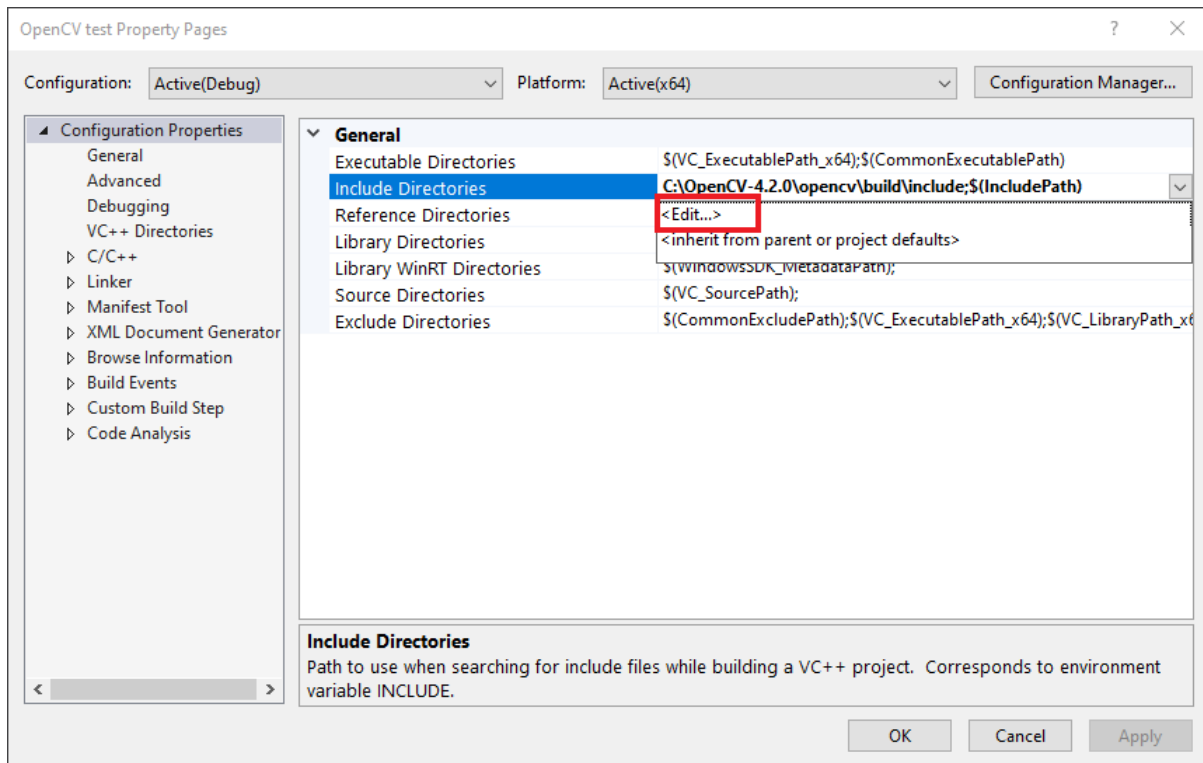
Change the target solution platform, and head to project properties

There are a bunch of properties to edit before being able to execute any code:

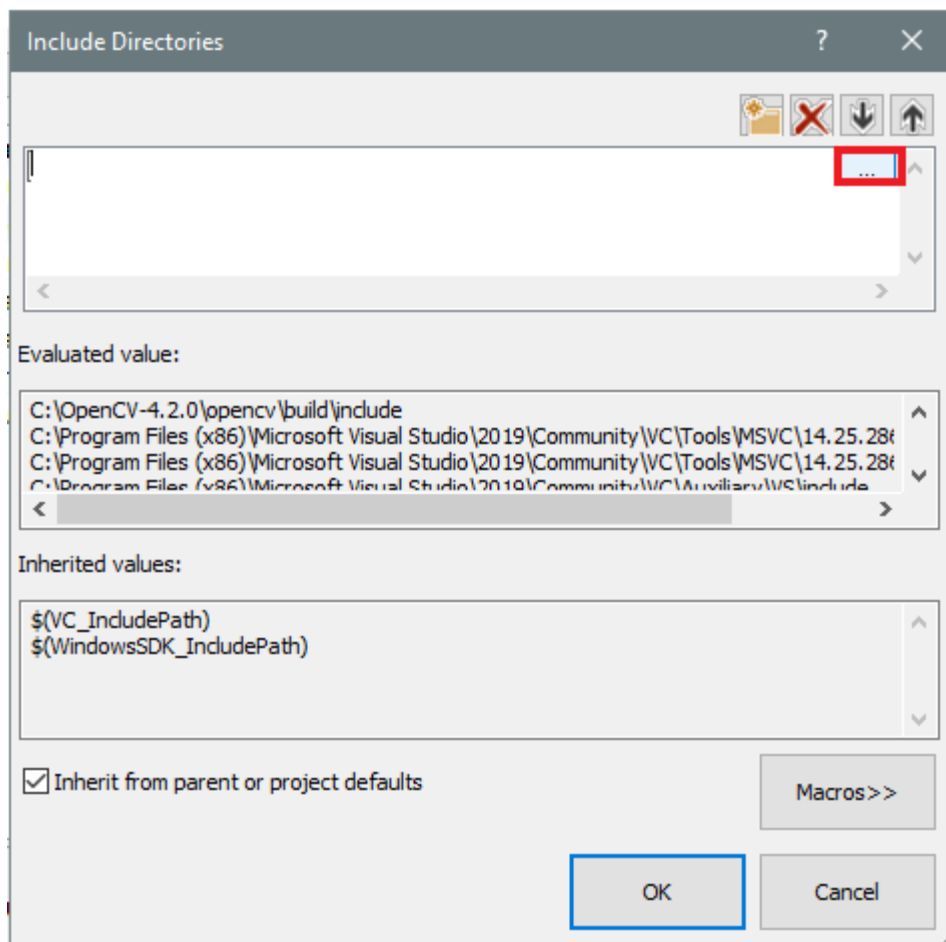
1. Go to **Configuration Properties/VC++ Directories** to add the include and library directories for OpenCV.



Edit both Include and Library Directories

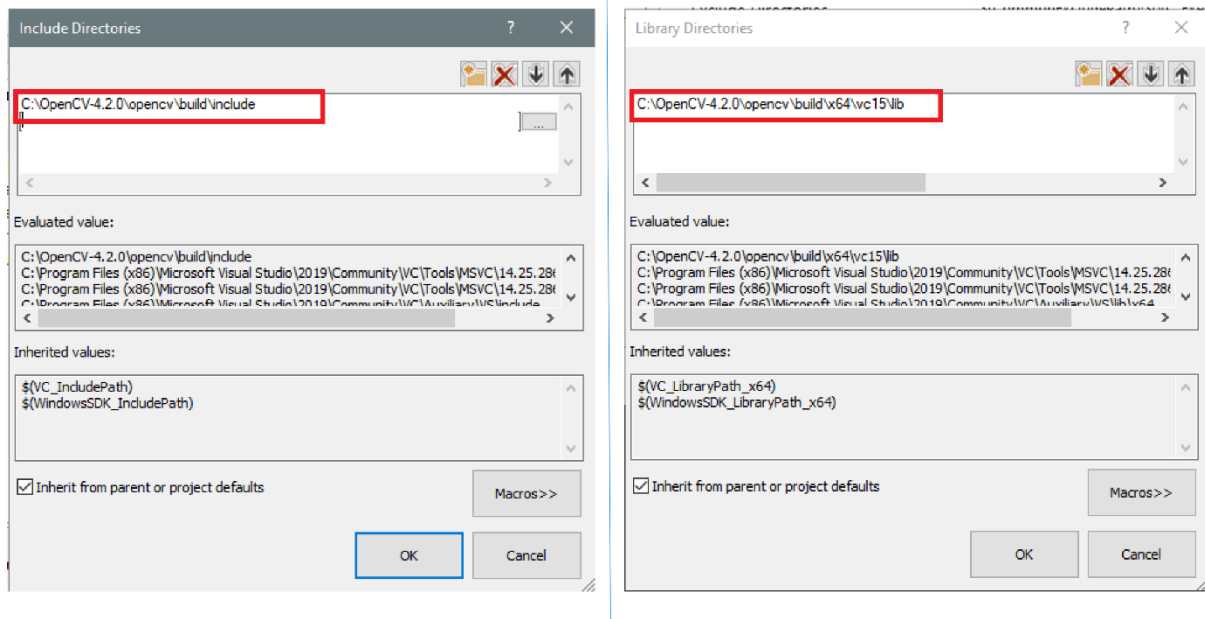


Click on the right arrow next to blue entry



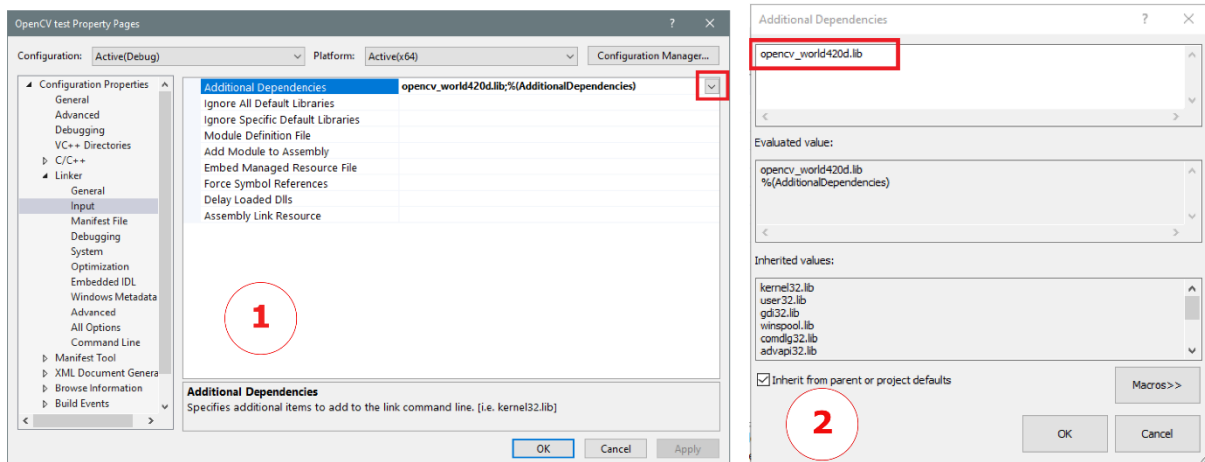
Include the internal system path

For the Include directory, you have to add the following path: C:\OpenCV-4.2.0\opencv\build\include. Do the same for the **Library Directories** adding this internal path: C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib.



Include and Lib directories both added to the project

2. Edit the VC++ project linker with the opencv_world420d.lib OpenCV dynamic library. You will find the **DLL (Dynamic Link Library)** here: C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib copy the name of the file opencv_world420d.lib and paste it in the dependency box.



Click the OK button when finished

Final Step: Test a bit of code

Finally, you can run this demo code in your visual studio IDE to see if it's all working fine.



The resulting image

And that's all, you can now use OpenCV with ease. 😊

Approach:

1. Find out a cell which is empty.
 - Keep track of Column Number and Row Number.
2. Consider A number to be filled in the empty cell.
 - First check if it is Safe (Traverse and check rows and columns do not have, selected element that we want to enter).
 - If Yes (Safe) assign the number to the current empty cell.
 - Else Try a different number.
3. Similarly try to fill other empty cells, Recursively.
4. If you are able to fill all the cells without any conflict then simply return the solved sudoku.

Algorithm:

Step 1: Start

Step 2: Check if Sudoku grid is Valid

Step 3: Create a function to check if a value is placed in current position in grid, does it becomes Unsafe or Not.

Step 4: Create a recursive function that takes a grid, to solve grid.

Step 5: Check for unassigned location.

- assign a number from 1 to 9
- check if assigning the number to current index makes the grid unsafe or not
- if safe then recursively call the function for all safe cases from 0 to 9.
- if any recursive call returns true, end the loop and return true. If no recursive call returns true then return false.

Step 6: If solution found Print Solved Sudoku

Else Print Invalid Sudoku, and Exit

Step 7: Stop

Functions Used:

1. `bool FindUnassignedLocation(int grid[N][N], int row, int col)`
- This function finds the location/Position of empty cells in the Grid.
2. `bool isSafe(int grid[N][N], int row, int col, int num);`
- Function to check if it is safe to, Enter the selected value at current position.
3. `bool SolveSudoku(int grid[N][N]);`
- Recursive Function to solve the Sudoku problem by putting different values in Empty Cells.
4. `bool UsedInRow(int grid[N][N], int row, int num);`
- Function to Check if Selected Value exists in the Row.
5. `bool UsedInCol(int grid[N][N], int col, int num);`
- Function to Check if Selected Value exists in the Column.
6. `bool UsedInBox(int grid[N][N], int boxStartRow, int boxStartCol, int num);`
- Function to Check if Selected Value exists in the 3x3 sub-grid.
7. `void printGrid(int grid[N][N]);`
- A utility function to print grid.
8. `void is_sudoku_valid(int grid[N][N]);`
- Function to check if Sudoku is Valid.
9. `void sudokuInput(int choice);`
- Input Function having 2 suduko grids: default and custom.
10. `int main();`
- Driver Function to run the Program.

Sample Input:

```
{ {5, 3, 0, 0, 7, 0, 0, 0, 0},  
  {6, 0, 0, 1, 9, 5, 0, 0, 0},  
  {0, 9, 8, 0, 0, 0, 0, 6, 0},  
  {8, 0, 0, 0, 6, 0, 0, 0, 3},  
  {4, 0, 0, 8, 5, 3, 0, 0, 1},  
  {7, 0, 0, 0, 2, 0, 0, 0, 6},  
  {0, 6, 0, 0, 0, 0, 2, 8, 0},  
  {0, 0, 0, 4, 1, 9, 0, 0, 5},  
  {0, 0, 0, 0, 8, 0, 0, 7, 9} };
```

Sample Output:

```
*****Enter Your Choice*****  
1: Default Sudoku Values  
2: Custom Sudoku Values  
Choice: 1  
  
***** Solved Sudoku *****  
  
5 3 4 | 6 7 8 | 9 1 2  
6 7 2 | 1 9 5 | 3 4 8  
1 9 8 | 3 4 2 | 5 6 7  
-----  
8 5 9 | 7 6 1 | 4 2 3  
4 2 6 | 8 5 3 | 7 9 1  
7 1 3 | 9 2 4 | 8 5 6  
-----  
9 6 1 | 5 3 7 | 2 8 4  
2 8 7 | 4 1 9 | 6 3 5  
3 4 5 | 2 8 6 | 1 7 9
```

Thank-you

Project 2: Sudoku Solver

Course: B.TECH CSE in Machine Learning and AI.

Student ID: 20011001

College: Graphic Era Hill University (Dehradun Campus)

Submitted By: Maskar Vishal