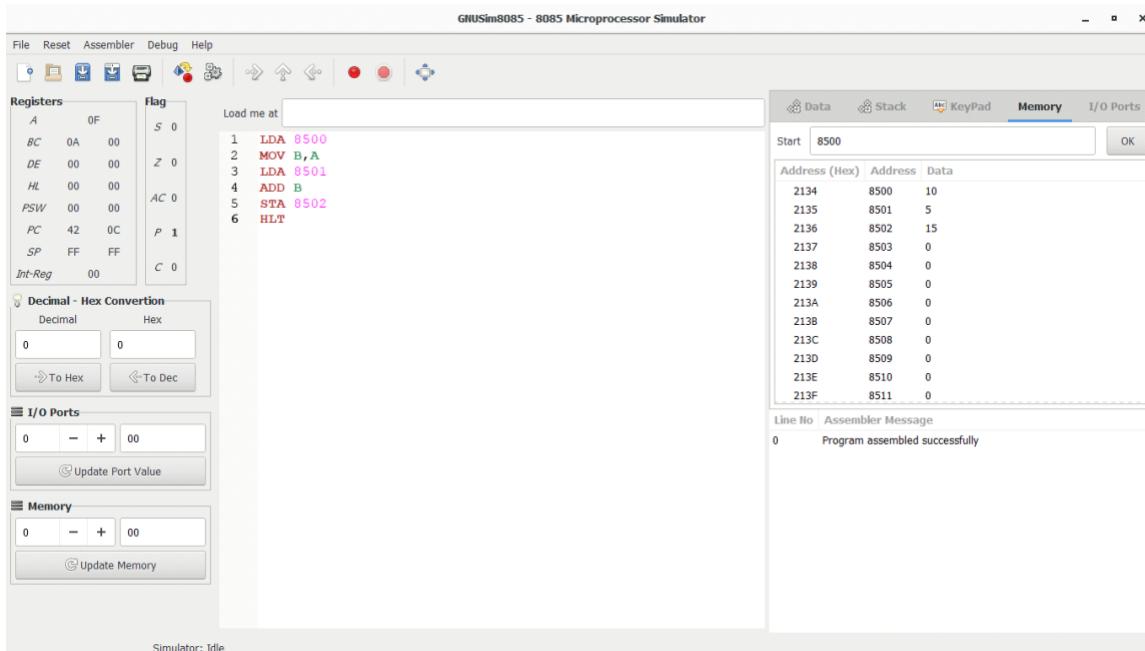


NAME : SRAVAN A K

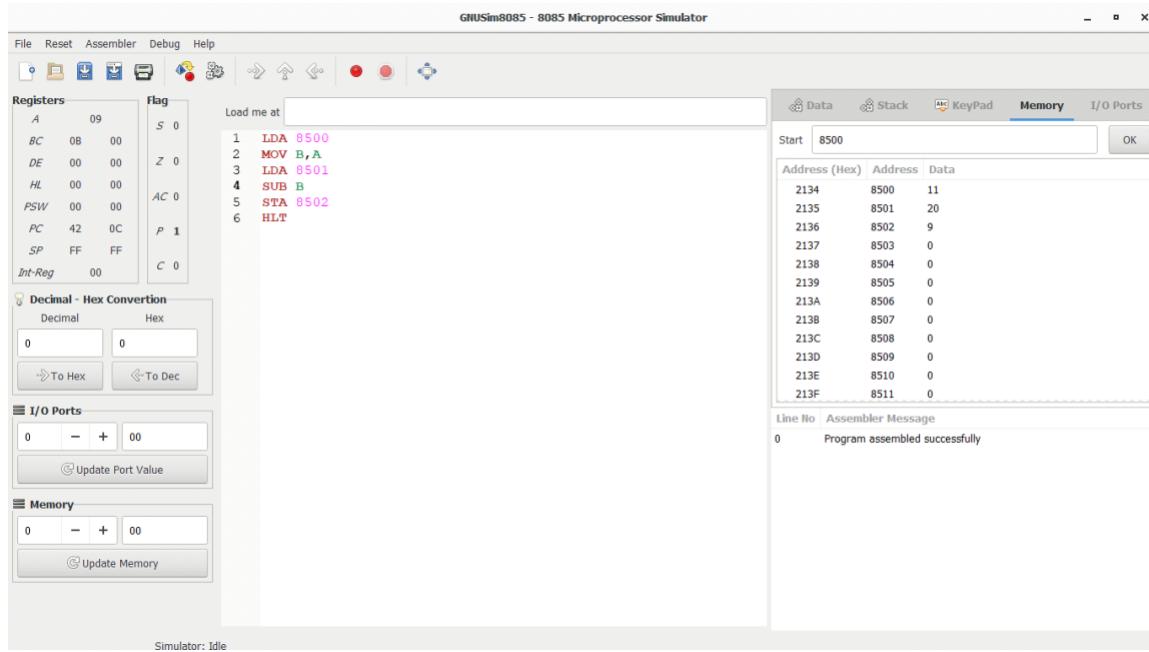
SUBJECT CODE AND NAME : CSA-1288 COMPUTER ARCHITECTURE FOR EMBEDDED SYSTEM

REG. NO. : 192110317

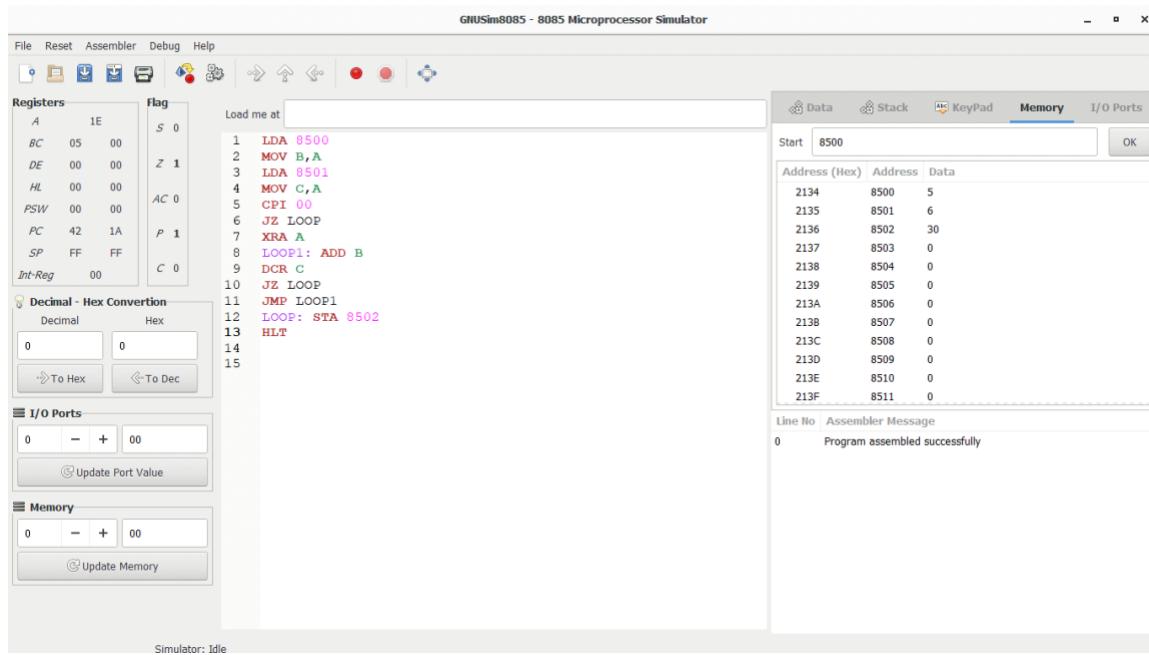
1. Write an assembly language program for adding two 8-bit data A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀ and B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ using 8085 processor



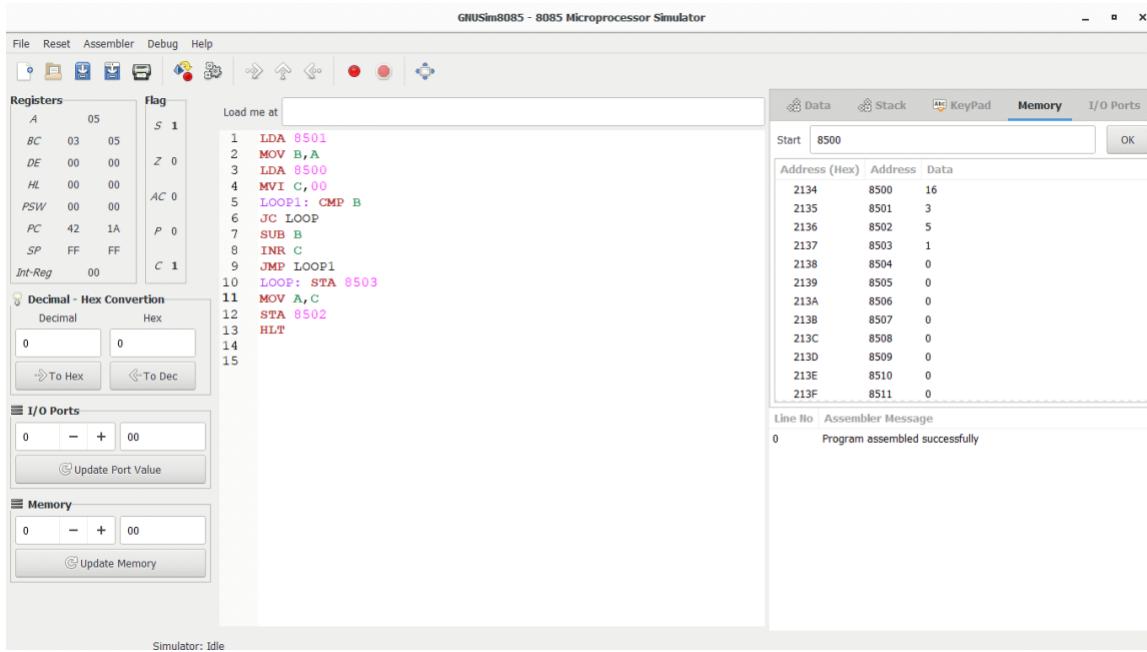
2. Write an assembly language program for subtraction of two 8-bit data A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀ and B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ using 8085 processor



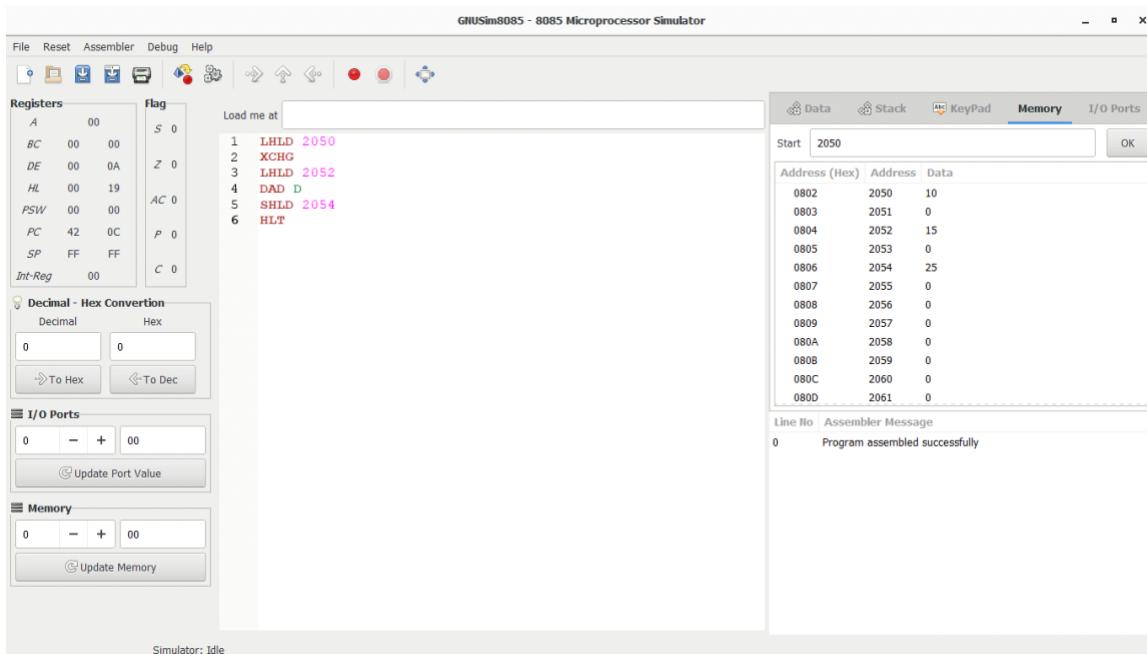
3. Write an assembly language program for multiplication of two 8-bit data A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀ and B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ using 8085 processor



4. Write an assembly language program for division of two 8-bit data A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀ and B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ using 8085 processor



5. Write an assembly language program for adding two 16-bit data using 8086 processor.



6. Write an assembly language program for subtracting two 16-bit data using 8086 processor.

GIUSim8085 - 8085 Microprocessor Simulator

Registers

A	00
BC	00 00
DE	00 28
HL	00 19
PSW	00 00
PC	42 14
SP	FF FF
Int-Reg	00

Flag

S	0
Z	1
AC	0
P	1
C	0

Load me at

```

1 LHLD 2050
2 XCHG
3 LHLD 2052
4 MVI C,00
5 MOV A,E
6 SUB L
7 STA 2054
8 MOV A,D
9 SBB H
10 STA 2055
11 HLT

```

Data Stack KeyPad Memory I/O Ports

Start 2050 OK

Address (Hex)	Address	Data
0802	2050	40
0803	2051	0
0804	2052	25
0805	2053	0
0806	2054	15
0807	2055	0
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No Assembler Message
0 Program assembled successfully

Simulator: Idle

7. Write an assembly language program for multiplying two 16-bit data using 8086 processor.

GIUSim8085 - 8085 Microprocessor Simulator

Registers

A	00
BC	00 00
DE	00 00
HL	00 00
PSW	00 00
PC	42 22
SP	00 0F
Int-Reg	00

Flag

S	0
Z	1
AC	0
P	1
C	0

Load me at

```

1 LHLD 2050
2 SPHL
3 LHLD 2052
4 XCHG
5 LXI H,0000H
6 LXI B,0000H
7 LOOP1: DAD SP
8 JNC LOOP2
9 INX B
10 LOOP2: DCX D
11 MOV A,E
12 ORA D
13 JNZ LOOP1
14 SHLD 2054
15 MOV L,C
16 MOV H,B
17 SHLD 2056
18 HLT

```

Data Stack KeyPad Memory I/O Ports

Start 2050 OK

Address (Hex)	Address	Data
0802	2050	15
0803	2051	0
0804	2052	5
0805	2053	0
0806	2054	75
0807	2055	0
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No Assembler Message
0 Program assembled successfully

Simulator: Idle

8. Write an assembly language program for dividing two 16-bit data using 8086 processor.

GRUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	05
BC	03 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 1A
SP	FF FF
Int-Reg	00

Flag

S	1
Z	0
AC	0
P	0
C	1

Load me at:

```

1 LDA 8501
2 MOV B,A
3 LDA 8500
4 MVI C,00
5 LOOP1: CMP B
6 JC LOOP
7 SUB B
8 INR C
9 JMP LOOP1
10 LOOP: STA 8503
11 MOV A,C
12 STA 8502
13 HLT
14
15

```

Data Stack KeyPad Memory I/O Ports

Start: 8500 OK

Address (Hex)	Address	Data
2134	8500	16
2135	8501	3
2136	8502	5
2137	8503	1
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

Line No Assembler Message

0 Program assembled successfully

I/O Ports

Memory

Simulator: Idle

9. Write an assembly language program to swap two 8-bit data using 8085 processor

BEFORE SWAP

GRUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	05
BC	03 05
DE	00 00
HL	00 00
PSW	00 00
PC	42 1A
SP	FF FF
Int-Reg	00

Flag

S	1
Z	0
AC	0
P	0
C	1

Load me at:

```

1 LDA 8085
2 MOV B,A
3 LDA 8086
4 STA 8085
5 MOV A,B
6 STA 8086
7 HLT
8

```

Data Stack KeyPad Memory I/O Ports

Start: 8085 OK

Address (Hex)	Address	Data
1F95	8085	45
1F96	8086	100
1F97	8087	0
1F98	8088	0
1F99	8089	0
1F9A	8090	0
1F9B	8091	0
1F9C	8092	0
1F9D	8093	0
1F9E	8094	0
1F9F	8095	0
1FA0	8096	0

Line No Assembler Message

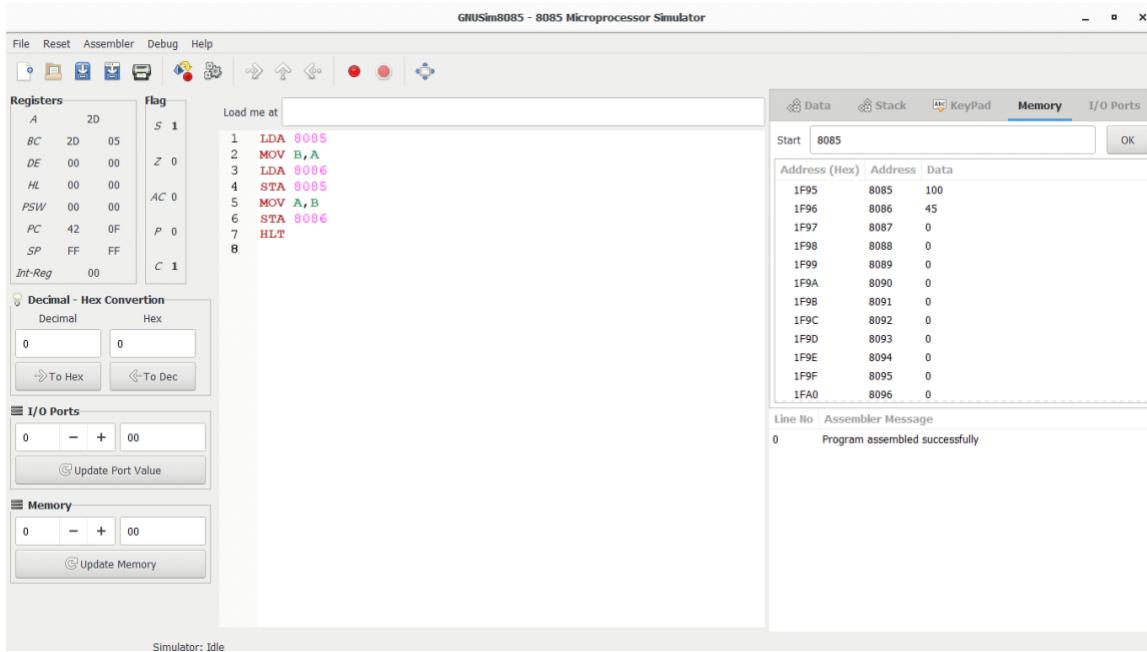
0 Program assembled successfully

I/O Ports

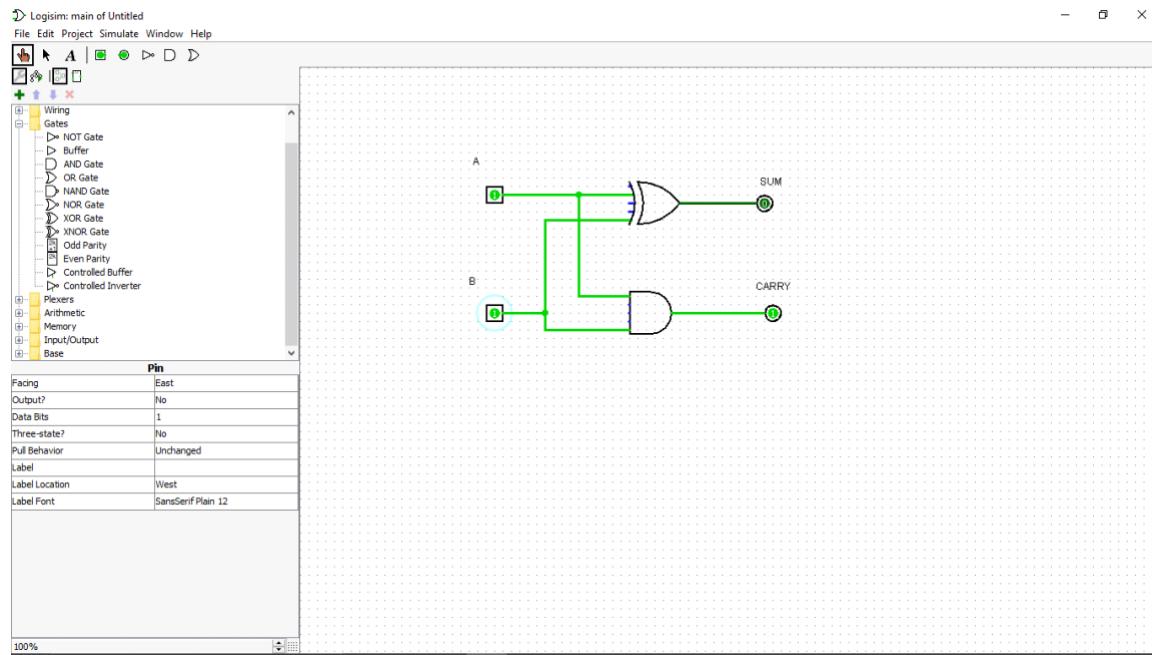
Memory

Simulator: Idle

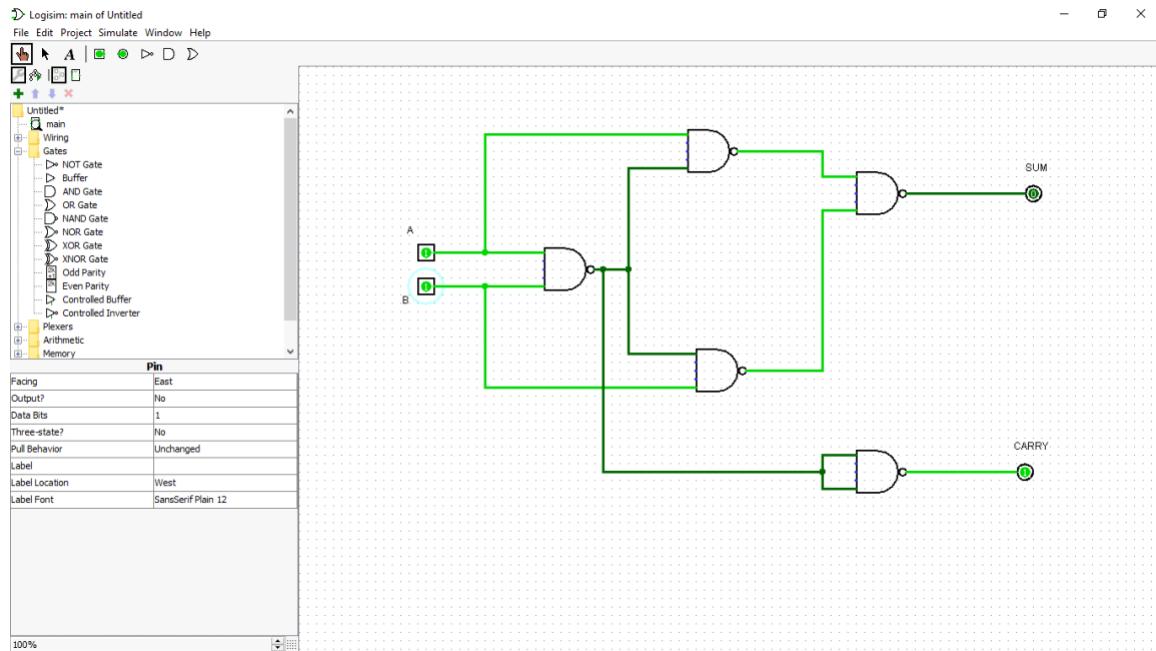
AFTER SWAP



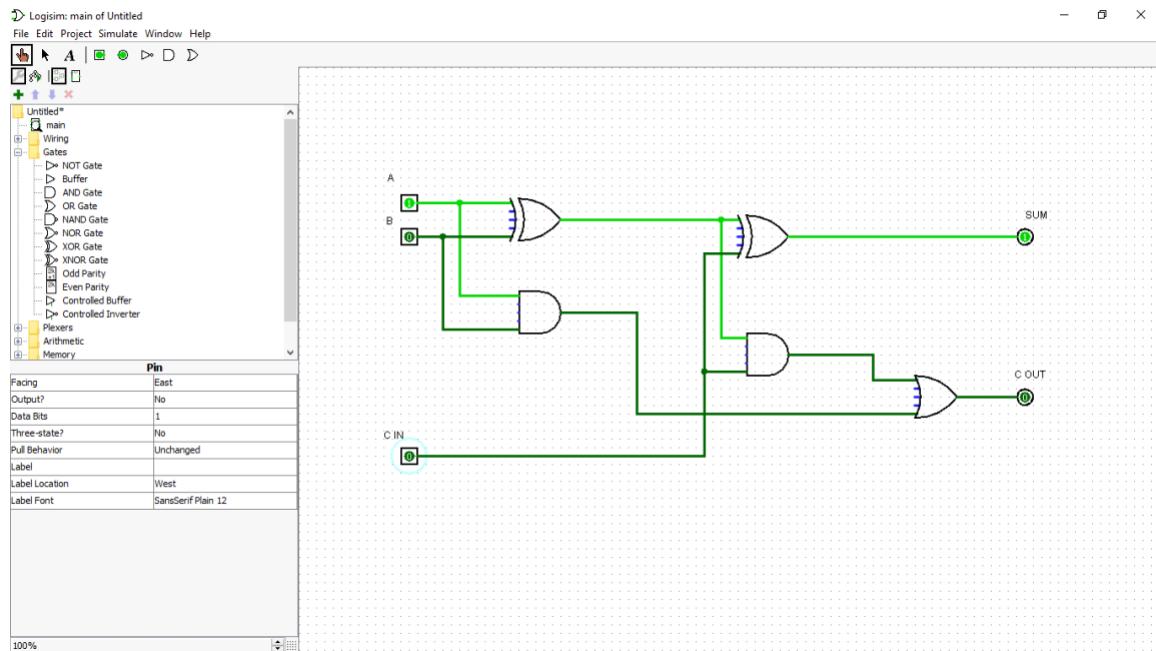
10. Design and implement 2-bit half adder using logisim simulator



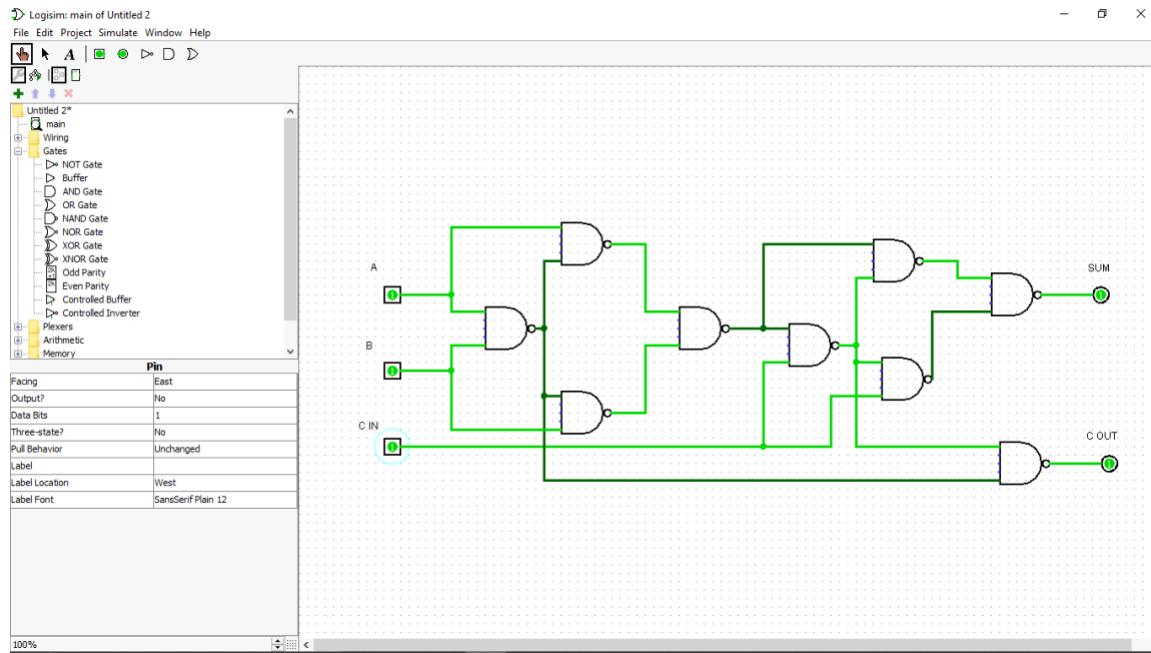
11. Design and implement 2-bit half adder with NAND using logisim simulator



12.Design and implement 3-bit full adder using logisim simulator.



13..Design and implement 3-bit full adder with NAND using logisim simulator



14. Write an assembly language program to find factorial of n in the given number.

```

#include <iostream>
using namespace std;
int main()
{
    int i, fact=1, number;
    cout<<"Enter any Number: ";
    cin>>number;
    for(i=1;i<=number;i++)
        fact=fact*i;
    cout<<"Factorial of " <<number<<" is: "<<fact<<endl;
    return 0;
}

```

15. Write an assembly language program to find the largest number in an array

```

C:\Users\Sakthivel G\Documents\CA\C++\largest no. in an array.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug
largest no. in an array.cpp
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int arr[100], tot, larg, i;
6     cout<<"Enter the Size (max. 100): ";
7     cin>>tot;
8     cout<<"Enter "<<tot<<" Array Elements: ";
9     for(i=0; i<tot; i++)
10    {
11        cin>>arr[i];
12        larg = arr[0];
13        for(i=1; i<tot; i++)
14        {
15            if(larg<arr[i])
16                larg = arr[i];
17        }
18        cout<<"\nLargest Number = "<<larg;
19        cout<<endl;
20    }
21

```

Enter the Size (max. 100): 5
Enter 5 Array Elements: 4
5
6
7
8
Largest Number = 8
Process exited after 11.66 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 21 Col: 1 Sel: 0 Lines: 21 Length: 432 Insert Done parsing in 0.015 seconds

16. Write a program to perform Booth's multiplication of two signed numbers using any high level language

```

booth mul.py - C:/Users/Sakthivel G/Documents/CA/booth mul.py (3.10.7)
File Edit Format Run Options Window Help
def add(ac, x, qrn):
    c = 0
    for i in range(qrn):
        ac[i] = ac[i] + x[i] + c;
        if (ac[i] > 1):
            ac[i] = ac[i] % 2
            c = 1
        else:
            c = 0
def complement(a, n):
    x = [0] * n
    x[0] = 1
    for i in range(n):
        a[i] = (a[i] + 1) % 2
    add(a, x, n)
def rightShift(ac, qr, qrn, qrnr):
    temp = ac[0]
    qr = qr[0]
    print("\t\trightShift\t", end = "")
    for i in range(qrn - 1):
        ac[i] = ac[i + 1]
        qr[i] = qr[i + 1]
    qr[qrn - 1] = temp
def display(ac, qr, qrn):
    for i in range(qrn - 1, -1, -1):
        print(ac[i], end = '')
    print("\t", end = "")
    for i in range(qrn - 1, -1, -1):
        print(qr[i], end = '')
def boothAlgorithm(br, ar, mt, qrnr, sc):

```

File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:/Users/Sakthivel G/Documents/CA/booth mul.py =====
q[n+1] BR AC QR sc
initial 0000 1010 4
0 0 rightShift 0000 0101 3
1 A = A - 0101
rightShift 1101 0010 2
0 0 rightShift 1110 1001 1
1 A = A + BR 0100
rightShift 0010 0100 0
Result = 0100
>>>

The code implements the Booth multiplication algorithm. It defines a function `main()` that takes two integers `br` and `qr` as input. It initializes arrays `mt` and `qrn`, and performs a series of operations including reversing `br`, calculating the complement of `mt`, and applying the Booth algorithm's logic (adding or shifting) based on the current bit of `br`. The result is printed at the end.

```
print("\t", end = "")\n\n    sc -= 1\n    print("\t", sc, sep = "")\n\ndef main():\n\n    mt = [0] * 10\n\n    brn = 4\n\n    br = [ 0, 1, 1, 0 ]\n\n    for i in range(brn - 1, -1, -1):\n        mt[i] = br[i]\n\n    br.reverse()\n\n    complement(mt, brn)\n\n    qrn = 4\n\n    sc = qrn\n\n    qr = [ 1, 0, 1, 0 ]\n    qr.reverse()\n\n    boothAlgorithm(br, qr, mt, qrn, sc)\n\n    print("\nResult = ", end = "")\n\n    for i in range(qrn - 1, -1, -1):\n        print(qr[i], end = "")\n    print()\n\nmain()
```

The IDLE Shell window shows the execution of the script and the resulting binary output:

```
File Edit Shell Debug Options Window Help\nPython 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32\nType "help", "copyright", "credits" or "license()" for more information.\n>>> ===== RESTART: C:/Users/Sakthivel G/Documents/GA/booth mul.py ======\nqn      q[n+1]          initial      AC      QR      sc\n0       0               0000      1010      4\n0       0               rightShift  0000      0101      3\n1       0               A = A - BR   1010\n0       0               rightShift  1101      0010      2\n1       0               rightShift  1110      1001      1\n1       0               A = A + BR   0100\n0       0               rightShift  0010      0100      0\n\nResult = 0100\n>>> |
```

17. Write a program to perform Restoring Division of two numbers using any high level language.

```

restoring division.py - C:/Users/Sakthivel G/Documents/CA/restoring division.py (3.10.7)
File Edit Format Run Options Window Help
def add(A, M):
    carry = 0
    Sum = ''
    for i in range (len(A)-1, -1, -1):
        temp = int(A[i]) + int(M[i]) + carry
        if (temp>1):
            Sum += str(temp % 2)
            carry = 1
        else:
            Sum += str(temp)
            carry = 0
    return Sum[::-1]
def compliment(m):
    M = ''
    for i in range (0, len(m)):
        M += str(int(m[i]) + 1) % 2
    M = add(M, '0001')
    return M

def restoringDivision(Q, M, A):
    count = len(M)

    print ('Initial Values: A:', A,
          ' Q:', Q, ' M:', M)

    while (count):
        print ("\nstep:", len(M)-count + 1, end = '')
        print (' Left Shift and Subtract: ', end = '')
        A = A[1:] + Q[0]

        comp_M = compliment(M)

        A = add(A, comp_M)
        print('A:', A)
        print('A:', A, ' Q:', Q[1:]+'_', end = '')
        if (A[0] == '1'):

            Q = Q[1:] + '0'
            print (' -Unsuccessful')

            A = add(A, M)
            print ('A:', A, ' Q:', Q, ' -Restoration')

        else:
            Q = Q[1:] + '1'
            print (' Successful')

            print ('A:', A, ' Q:',
                  Q, ' -No Restoration')
        count -= 1

    print ('\nQuotient(Q):', Q,
          ' Remainder(A):', A)

if __name__ == "__main__":
    dividend = '0110'
    divisor = '0100'

    accumulator = '0' * len(dividend)

    restoringDivision(dividend, divisor, accumulator)

```

IDLE Shell 3.10.7

```

Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:/Users/Sakthivel G/Documents/CA/restoring division.py =====
Initial Values: A: 0000 Q: 0110 M: 0100

step: 1 Left Shift and Subtract: A: 1100
A: 1100 Q: 110 -Unsuccessful
A: 0000 Q: 1100 -Restoration

step: 2 Left Shift and Subtract: A: 1101
A: 1101 Q: 100 -Unsuccessful
A: 0001 Q: 1000 -Restoration

step: 3 Left Shift and Subtract: A: 1111
A: 1111 Q: 000 -Unsuccessful
A: 0011 Q: 0000 -Restoration

step: 4 Left Shift and Subtract: A: 0010
A: 0010 Q: 000 Successful
A: 0010 Q: 0001 -No Restoration

Quotient(Q): 0001 Remainder(A): 0010
>>>

```

18. Write a program to find the Hit ratio for the given number of Hits and Misses in Cache memory using any high level language

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a C program named Untitled1.cpp. The code prompts the user for the total number of cache hits and misses, calculates the cache hit ratio, and prints it. On the right, the terminal window shows the execution of the program, inputting values 43 and 11, and displaying the results.

```

C:\Users\Sakthivel G\Documents\CA\C++\Untitled1.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug Untitled1.cpp
1 #include <stdio.h>
2 int main()
3 {
4     float cachehit, cachemiss;
5     printf("\nEnter the total number of cache hits:");
6     scanf("%d", &cachehit);
7     printf("\nEnter the number of cache misses:");
8     scanf("%d", &cachemiss);
9     cachehitratio=cachehit/(cachehit+cachemiss);
10    printf("\n Cache Hit Ratio: %f", cachehitratio);
11    printf("\n Cache Miss Ratio: %f", 1-cachehitratio);
12    return 0;
13 }
14

```

C:\Users\Sakthivel G\Documents\CA\C++\Untitled1.exe
enter the total number of cache hits:43
enter the number of cache misses:11
Cache Hit Ratio: 0.796296
Cache Miss Ratio: 0.203704
Process exited after 26.25 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Sakthivel G\Documents\CA\C++\Untitled1.exe
- Output Size: 128.6015625 KB
- Compilation Time: 2.09s

Line: 14 Col: 1 Sel: 0 Lines: 14 Length: 408 Insert Done parsing in 0.157 seconds

19. Write a program to convert Decimal number to Binary number using any high level language

The screenshot shows two windows of the Python IDLE environment. The left window is a code editor for a file named bin to dec.py, containing a function to convert a decimal number to binary. The right window is a shell window showing the execution of the script, where the user inputs a decimal value and the script outputs its binary representation.

```

bin to dec.py - C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/bin to dec...
File Edit Format Run Options Window Help
def DecimalToBinary(num):
    if num >= 1:
        DecimalToBinary(num // 2)
        print(num % 2, end = '')
if __name__ == '__main__':
    dec_val = 24
    DecimalToBinary(dec_val)

```

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/bin to dec.py
011000
>>>

20. Write a program to convert Decimal number to an Octal number using any high level language

The screenshot shows a Python IDE interface with two windows. The left window is titled 'conversion.py' and contains the following code:

```

File Edit Format Run Options Window Help
def decToOctal(n):
    octalNum = [0] * 100
    i = 0
    while (n != 0):
        octalNum[i] = n % 8
        n = int(n / 8)
        i += 1
    for j in range(i - 1, -1, -1):
        print(octalNum[j], end="")
n = 33
decToOctal(n)

```

The right window is titled 'IDLE Shell 3.10.7' and shows the output of running the program:

```

File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/conversi
on.py
41
>>>

```

21. Write a program to convert Binary number to Decimal number using any high level language.

The screenshot shows a Python IDE interface with two windows. The left window is titled 'conversion.py' and contains the following code:

```

File Edit Format Run Options Window Help
def binaryToDecimal(binary):
    binary = binary
    decimal, i, n = 0, 0, 0
    while(binary != 0):
        dec = binary % 10
        decimal = decimal + dec * pow(2, i)
        binary = binary//10
        i += 1
    print(decimal)
if __name__ == '__main__':
    binaryToDecimal(100)
    binaryToDecimal(101)
    binaryToDecimal(1001)

```

The right window is titled 'IDLE Shell 3.10.7' and shows the output of running the program:

```

File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/conversi
on.py
4
5
9
>>>

```

22. Write a program to find the CPU performance of a processor using any high level language.

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named Untitled1.cpp with the following content:

```

1 #include <stdio.h>
2 int main()
3 {
4     float cr;
5     int p,pl,i;
6     float ct,max;
7     int n=1000;
8     for(i=0;i<n;i++)
9     {
10        cpu[5]=0;
11    }
12    printf("\nEnter the number of processors:");
13    scanf("%d",&p);
14    pl=p;
15    for(i=0;i<p;i++)
16    {
17        printf("\nEnter the Cycles per Instruction of processor:");
18        scanf("%f",&cp);
19        printf("\nEnter the clockrate in GHz:");
20        scanf("%f",&cr);
21        scanf("%f",&ct);
22        ct=1000*cp/cr;
23        printf("The CPU time is: %f",ct);
24        cpu[i]=ct;
25    }
26    max=cpu[0];
27    //printf("%f", max);
28    for(i=0;i<p;i++)
29    {

```

To the right, a terminal window titled "C:\Users\Sakthivel G\Documents\CA\C++\Untitled1.exe" shows the output of the program:

```

Enter the number of processors:1
Enter the Cycles per Instruction of processor:1.5
Enter the clockrate in GHz:3
The CPU time is: 500.000000
The processor has lowest Execution time is: 500.000000
Process exited after 26.8 seconds with return value 0
Press any key to continue . .

```

Below the code editor, the compiler log window shows the compilation results:

```

Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Sakthivel G\Documents\CA\C++\Untitled1.exe
- Output Size: 128.6015625 KB
- Compilation Time: 0.45s

```

At the bottom, status bar information includes: Line: 35, Col: 2, Sel: 0, Lines: 35, Length: 639, Insert, Done parsing in 0 seconds.

23. Write an assembly language program to find 1's and 2's complement of 8 bit number.

Memory	Mnemonics	Operands	Comment
2000	LDA	[3000]	[A] <- [3000]
2003	CMA		[A] <- [A^]
2004	STA	[3001]	1's complement
2007	ADI	01	[A] <- [A] + 01
2009	STA	[3002]	2's complement
200C	HLT		Stop

24. Design of 2 stage pipeline for addition and subtraction of two numbers using any high level language

program:

```
counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1
b=int(input("ENTER NUMBER-2-"))
counter=counter+1
print("1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION")

print("Enter Your Choice")
choice=int(input())
if choice==1:

    print("Performing Addition...")
    res=a+b
    counter=counter+1

if choice==2:
    print("Performing Subtraction...")
    res=a-b
    counter=counter+1

if choice==3:
    print("Performing Multiplication")
    res=a*b
    counter=counter+1

if choice==4:
    if b==0:
        print("Denominator can't be Zero")
    print("Performing Division")
    res=a/b
    counter=counter+1

if choice>=5:
    print("Enter Correct Input")

print(res)
```

```

counter=counter+1
print("CYCLE VALUE IS:",counter)
ins=int(input("Enter the No.of instructions:"))
performance_measure =ins/counter
print("performance measure is:" performance_measure)

```

OUTPUT:

```

IDE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2-rc1-0.g4d8e7c0, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\CA.py =====
>>> ENTER NUMBER-1-2
>>> ENTER NUMBER-2-18
>>> 1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION
>>> Enter Your Choice
>>> 1
>>> Performing Addition...
>>> 41
>>> CYCLE VALUE IS: 1
>>> Enter the No.of instructions:4
>>> performance measure is: 1.0
>>>

```

25.Design of 3 stage pipeline for AND, OR, NAND of two numbers using any high level language.

program:

```

counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1
b=int(input("ENTER NUMBER-2-"))
counter=counter+1
res= a and b
counter=counter+1
print(res)
counter=counter+2
INS=int(input("enter no. of instructions:"))
performance_measure=INS/counter
print("performance measure:",performance_measure)

```

OUTPUT:



The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The status bar at the bottom right indicates Line 10 and Column 0. The code window displays the following session:

```
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\OR.py =====
ENTER NUMBER-1-1001
ENTER NUMBER-2-1010
1010
enter no. of instructions:4
performance measure: 0.8
>>>
```

26.Design of 4 stage pipeline for multiplication and division of two numbers using any high level language

program:

```
counter=1
```

```
a=int(input("ENTER NUMBER-1-"))
```

```
counter=counter+1
```

```
b=int(input("ENTER NUMBER-2-"))
```

```
counter=counter+1
```

```
print("1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION")
```

```
print("Enter Your Choice")
```

```
choice=int(input())
```

```
if choice==1:
```

```
    print("Performing Addition...")
```

```
    res=a+b
```

```
    counter=counter+1
```

```
if choice==2:
```

```
print("Performing Subtraction...")
res=a-b
counter=counter+1
if choice==3:
    print("Performing Multiplication")
    res=a*b
    counter=counter+1
if choice==4:
    if b==0:
        print("Denominator can't be Zero")
    print("Performing Division")
    res=a/b
    counter=counter+1
if choice>=5:
    print("Enter Correct Input")

print(res)
counter=counter+3
print("CYCLE VALUE IS:",counter)
ins=int(input("Enter the No.of instructions:"))
performance_measure =ins/counter
print("performance measure is:",performance_measure)
```

OUTPUT:

```

IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\CA.py =====
ENTER NUMBER-1-3
ENTER NUMBER-2-2
1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION
Enter Your Choice
2
Performing Subtraction...
11
CYCLE VALUE IS: 7
Enter the No.of instructions:4
performance measure is: 0.5714285714285714
>>>

```

Ln: 15 Col: 0

27. Write a program to convert Decimal number to Hexadecimal number using any high level language.

```

hugyu.py - C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/hugyu.py (3.10.7)
File Edit Format Run Options Window Help
conversion_table = {0: '0', 1: '1', 2: '2', 3: '3', 4: '4',
                    5: '5', 6: '6', 7: '7',
                    8: '8', 9: '9', 10: 'A', 11: 'B', 12: 'C',
                    13: 'D', 14: 'E', 15: 'F'}

def decimalToHexadecimal(decimal):
    hexadecimal = ''
    while(decimal > 0):
        remainder = decimal % 16
        hexadecimal = conversion_table[remainder] + hexadecimal
        decimal = decimal // 16

    return hexadecimal

decimal_number = 69
print("The hexadecimal form of", decimal_number,
      "is", decimalToHexadecimal(decimal_number))

```

Ln: 6

Ln: 2 Col: 3

28. Write a program to convert Binary to Octal number using any high level language.

```

huguyu.py - C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/huguyu.py (3.10.7)
File Edit Format Run Options Window Help
def createMap(bin_oct_map):
    bin_oct_map["000"] = '0'
    bin_oct_map["001"] = '1'
    bin_oct_map["010"] = '2'
    bin_oct_map["011"] = '3'
    bin_oct_map["100"] = '4'
    bin_oct_map["101"] = '5'
    bin_oct_map["110"] = '6'
    bin_oct_map["111"] = '7'
def convertBinToOct(bin):
    l = len(bin)
    t = -1
    if '.' in bin:
        t = bin.index('.')
        len_left = t
    else:
        len_left = l
    for i in range(l, (3 - len_left % 3) % 3 + 1):
        bin = '0' + bin
    if (t != -1):
        len_right = l - len_left - 1
        for i in range(l, (3 - len_right % 3) % 3 + 1):
            bin = bin + '0'
    bin_oct_map = {}
    createMap(bin_oct_map)
    i = 0
    octal = ""
    while (True):
        octal += bin_oct_map[bin[i:i + 3]]
        i += 3
        if (i == len(bin)):
            break
        if (bin[i] == '.'):
            octal += '.'
            i += 1
    return octal
bin = "111001010010100001.010110110011011"
print("Octal number = ", convertBinToOct(bin))

```

29. Write an assembly language program to find the Greatest of 2 numbers

```

C:\Users\Sakthivel G\Documents\CA\C++\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int numOne, numTwo, larg;
6     cout<<"Enter the Two Numbers: ";
7     cin>>numOne>>numTwo;
8     if(numOne>numTwo)
9         larg = numOne;
10    else
11        larg = numTwo;
12    cout<<"\nLargest = "<<larg;
13    cout<<endl;
14    return 0;
15

```

30. Write an assembly language program to find the Smallest of 2 numbers.

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C program named mlp.cpp. The code reads two integers from the user and prints the smaller one. The output window shows the execution of the program, where it asks for two numbers, receives 15 and 7, and then prints "Smallest of the two number is: 7". Below the code editor is a compilation results window showing no errors or warnings, and the output file is mlp.exe.

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     int a, b, small;
6     printf("Enter any two number: ");
7     scanf("%d%d", &a, &b);
8     if(a>b)
9         small=a;
10    else
11        small=b;
12    printf("\nSmallest of the two number is: %d", small);
13    getch();
14    return 0;
15

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

Abort Compilation

Shorten compiler paths

Line: 4 Col: 2 Sel: 0 Lines: 15 Length: 292 Insert Done parsing in 0.125 seconds

31. Write an assembly language program to find the LCM of 2 numbers.

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C program named mlp.cpp. The program calculates the Least Common Multiple (LCM) of two numbers by incrementing a third variable until both input numbers are divisible by it. The output window shows the execution of the program, which prints "The LCM of 7 and 5 is 35". Below the code editor is a compilation results window showing no errors or warnings, and the output file is mlp.exe.

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int a=7, b=5, lcm;
5     if(a>b)
6         lcm = a;
7     else
8         lcm = b;
9     while(1) {
10         if( lcm%a==0 && lcm%b==0 ) {
11             cout<<"The LCM of "<<a<<" and "<<b<<" is "<<lcm;
12             break;
13         }
14         lcm++;
15     }
16     return 0;
17 }

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

Abort Compilation

Shorten compiler paths

Line: 8 Col: 12 Sel: 0 Lines: 17 Length: 300 Insert Done parsing in 0.015s

32. Write an assembly language program to find the GCD of 2 numbers

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <iostream>
2 using namespace std;
3 int gcd(int a, int b) {
4     if (b == 0)
5         return a;
6     return gcd(b, a % b);
7 }
8 int main() {
9     int a = 105, b = 30;
10    cout << "GCD of " << a << " and " << b << " is " << gcd(a, b);
11    return 0;
12 }

```

Output window:

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.exe
GCD of 105 and 30 is 15
Process exited after 0.0876 seconds with return value 0
Press any key to continue . .

```

Compiler Log:

```

Compiler Resources Compile Log Debug Find Results Close
Compilation results...
Abort Compilation
Shorten compiler paths
Line: 1 Col: 8 Sel: 0 Lines: 12 Length: 243 Insert Done parsing in 0.015 seconds

```

33. Write an assembly language program to arrange numbers in Ascending order.

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <iostream>
2 using namespace std;
3 void swapNums(int num[], int len);
4 void sortNums(int num[], int first, int second);
5 int main()
6 {
7     int num[] = {1, 12, 8, 10};
8     int size_nums = sizeof(num)/sizeof(num[0]);
9     cout << "Before sorting the array is: ";
10    for (int i = 0; i < size_nums; i++)
11        cout << num[i] << " ";
12    cout << endl;
13    sortNums(num, size_nums);
14    cout << "After sorting the array is: ";
15    for (int i = 0; i < size_nums; i++)
16        cout << num[i] << " ";
17    cout << endl;
18    return 0;
19 }
20 void sortNums(int num[], int len)
21 {
22     bool isSwapped;
23     for (int i = 0; i < len; i++)
24     {
25         isSwapped = false;
26         for (int j = i; j < len - i; j++)
27         {
28             if (num[j] > num[j + 1])
29             {
30                 swapNums(num, j, j + 1);
31                 isSwapped = true;
32             }
33         }
34         if (!isSwapped)
35             break;
36     }
37 }
38 void swapNums(int num[], int first, int second)
39 {
40     int curr = num[first];
41     num[first] = num[second];
42     num[second] = curr;
43 }

```

Output window:

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.exe
Before sorting the array is:
1 12 6 8 10
After sorting the array is:
6 8 10 12

```

Compiler Log:

```

Compiler Resources Compile Log Debug Find Results
Line: 46 Col: 2 Sel: 0 Lines: 54 Length: 906 Insert Done parsing in 0.016 seconds

```

34. Write an assembly language program to arrange numbers in descending order

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C++ program named mlp.cpp. The code defines an array arr with values { 3, 45, 54, 71, 76, 12 }, calculates its size, and then prints it in descending order. The output window shows the sorted array: 76 71 54 45 12 3. The compiler log window shows no errors or warnings, and the status bar indicates the code has been parsed in 0 seconds.

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main()
5 {
6     int arr[] = { 3, 45, 54, 71, 76, 12 };
7     int n = sizeof(arr) / sizeof(arr[0]);
8     cout << "Array: ";
9     for (int i = 0; i < n; i++)
10    | cout << arr[i] << " ";
11    sort(arr, arr + n, greater<int>());
12    cout << "InDescending Sorted Array:\n";
13    for (int i = 0; i < n; i++)
14    | cout << arr[i] << " ";
15    return 0;
16 }
17
18

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

Abort Compilation

Shorten compiler paths

Line: 1 Col: 1 Sek: 0 Lines: 18 Length: 378 Insert Done parsing in 0 seconds

35. Write an assembly language program to find the Minimum number in an array.

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C++ program named mlp.cpp. The code defines an array x with values { 2, 4, 6, 8, 0, 1 }, calculates its length, and then iterates through the array to find the smallest value. The output window shows the result: Smallest : 0. The compiler log window shows no errors or warnings, and the status bar indicates the code has been parsed in 0.016 seconds.

```

C:\Users\Sakthivel G\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int x[] = { 2, 4, 6, 8, 0, 1 };
7     int len = sizeof(x) / sizeof(x[0]);
8     if (len > 0) {
9         int smallest = x[0];
10
11        for (int i = 1; i < len; i++) {
12            if (smallest > x[i]) {
13                smallest = x[i];
14            }
15        }
16
17        cout << "Smallest : " << smallest << endl;
18    } else {
19        cout << "No elements in the array." << endl;
20    }
21

```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

Abort Compilation

Shorten compiler paths

Line: 21 Col: 1 Sek: 0 Lines: 21 Length: 487 Insert Done parsing in 0.016 seconds

36. Write an assembly language program to find 1's complement of 8 bit number

```

C:\Users\Sakthivel\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 unsigned int onesComplement(unsigned int n)
5 {
6     vector<int> v;
7     while (n != 0) {
8         v.push_back(n % 2);
9         n = n / 2;
10    }
11    reverse(v.begin(), v.end());
12    for (int i = 0; i < v.size(); i++) {
13        if (v[i] == 0)
14            v[i] = 1;
15        else
16            v[i] = 0;
17    }
18    int two = 1;
19    for (int i = v.size() - 1; i >= 0; i--) {
20        n = n + v[i] * two;
21        two = two * 2;
22    }
23    return n;
24}
25 int main()
26 {
27     unsigned int n = 22;
28     cout << onesComplement(n);
29     return 0;
30 }

```

Compiler Resources Compile Log Debug Find Results

Line: 25 Col: 1 Sel: 0 Lines: 31 Length: 506 Insert Done parsing in 0 seconds

37. Write an assembly language program to find 2's complement of 8 bit number

```

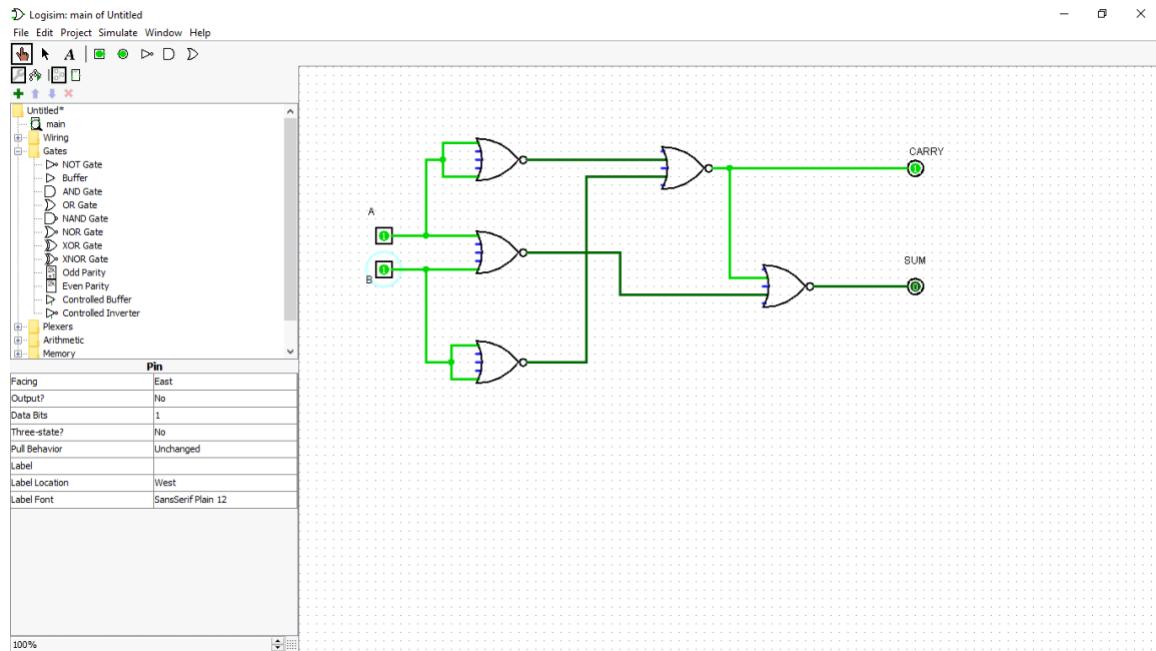
C:\Users\Sakthivel\Documents\CA\C+\mlp.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug mlp.cpp
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define SIZE 8
4 int main()
5 {
6     int i, carry = 1;
7     char num[SIZE + 1], one[SIZE + 1], two[SIZE + 1];
8     printf("Enter the binary number\n");
9     gets(num);
10    for(i = 0; i < SIZE; i++){
11        if(num[i] == '0'){
12            one[i] = '1';
13        }
14        else if(num[i] == '1'){
15            one[i] = '0';
16        }
17    }
18    one[SIZE] = '\0';
19    printf("One's complement of binary number %s is %s\n",num, one);
20    for(i = SIZE - 1; i >= 0; i--){
21        if(one[i] == '1' && carry == 1){
22            two[i] = '0';
23        }
24        else if(one[i] == '0' && carry == 1){
25            two[i] = '1';
26            carry = 0;
27        }
28        else{
29            two[i] = one[i];
30        }
31    }
32    two[SIZE] = '\0';
33    printf("Two's complement of binary number %s is %s\n",num, two);
34    return 0;

```

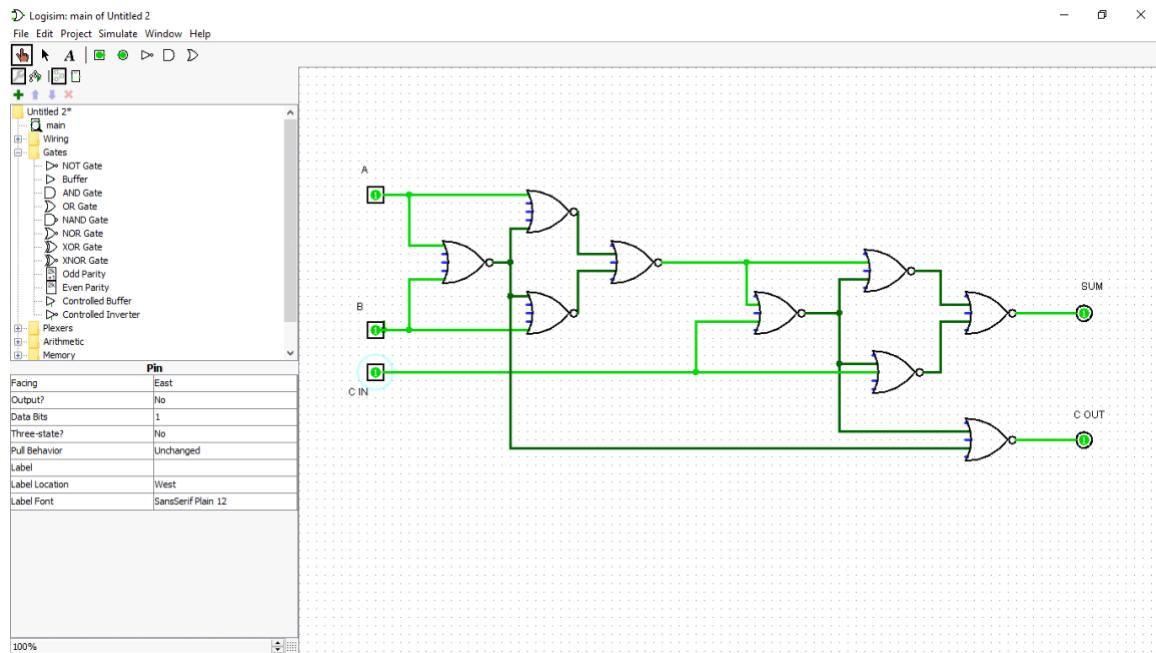
Compiler Resources Compile Log Debug Find Results

Line: 34 Col: 2 Sel: 0 Lines: 34 Length: 822 Insert Done parsing in 0.016 seconds

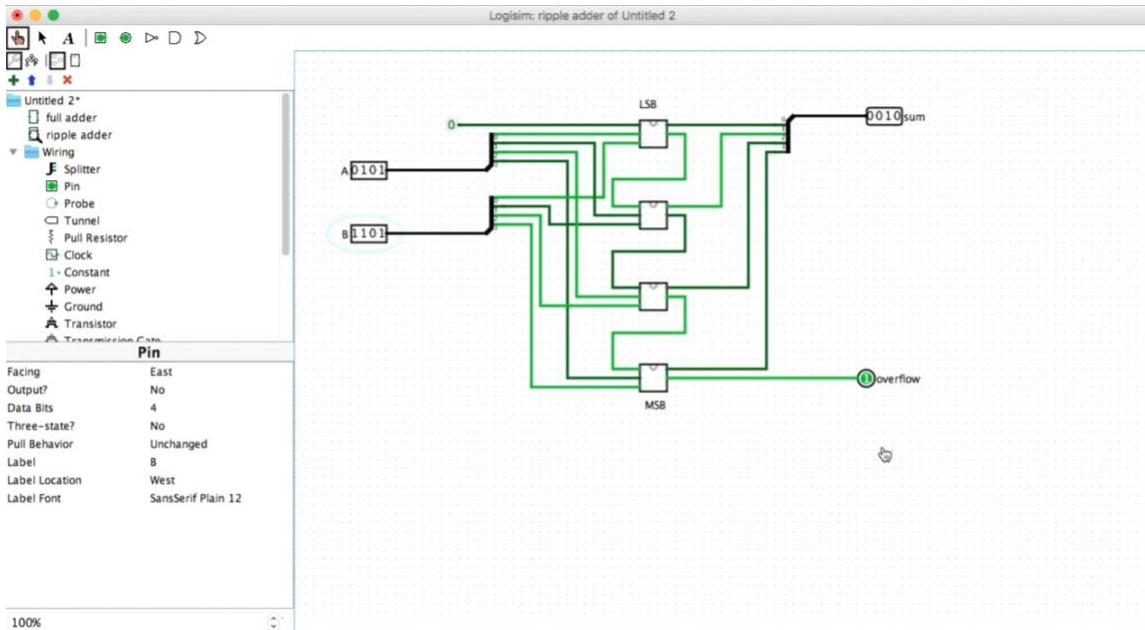
38. Design and implement 2-bit Half adder with NOR gates using logisim simulator



39.Design and implement 3-bit Full adder with NOR gates using logisim simulator



40.Design and implement 4-bit ripple carry adder circuit using logisim simulator



41. Write a program to convert Binary number to Hexadecimal number using any high level language.

```

hugyu.py - C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/hugyu.py (3.10.7)
File Edit Format Run Options Window Help
print("Enter the Binary Number: ")
bnum = int(input())
hex = 0
mul = 1
chk = 1
i = 0
hnum = []
while bnum!=0:
    rem = bnum%10
    hex = hex + (rem*mul)
    if chk%4==0:
        if hex>10:
            hex = hex+48
            val = chr(hex)
            hnum.insert(i, val)
        else:
            hex = hex+55
            val = chr(hex)
            hnum.insert(i, val)
    mul = 1
    hex = 0
    chk = 1
    i = i+1
else:
    mul = mul*2
    chk = chk+1
    bnum = int(bnum/10)
if chk==1:
    hex = hex+48
    val = chr(hex)
    hnum.insert(i, val)
if chk==1:
    i = i-1
print("\nEquivalent Hexadecimal Value = ", end="")
while i>=0:
    print(end=hnum[i])
    i = i-1
print()

```

File Edit Shell Debug Options Window Help
 Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
 Type "help", "copyright", "credits" or "license()" for more information.
 >>>
 RESTART: C:/Users/Sakthivel G/AppData/Local/Programs/Python/Python310/hugyu.py
 Enter the Binary Number:
 1001
 >>>
 Equivalent Hexadecimal Value = 9

42. Write an assembly language program to find the given 8-bit number is Prime or not using 8085 processor

Address	Mnemonics	Opcod e	LSB	MSB	Comments / Explanation
2000	LDA 3000	3A	00	30	Load accumulator
2003	MOV C,A	4F			Copy value of Reg. From Accumulator
2004	CPI 02H	FE	02		Compare Accumulator with 02
2006	JZ 2025	CA	25	20	If equal z=0 then jump to 2025
2009	CPI 01H	FE	01		Compare accumulator with 01
200B	JZ 202D	CA	2D	20	Jump to 202D
200E	CPI 01H	FE	01		Compare accumulator with 01
2010	JZ 202D	CA	2D	20	If equal jump to 202D
2013	MVI B,02H	06	02		B = 2
2015	SUB B	90			Subtract b from Accumulator
2016	JZ 202D	CA	2D	20	Jump to 202D
2019	JNC 2015	D2	15	20	If a=0 then jump to 2015
201C	INR B	04			Increment B
201D	MOV A,C	79			Copy C to A
201E	CMP B	B8			Compare B with Accumulator
201F	JZ 2025	CA	25	20	If zero jump to 2025
2022	JNZ 2015	CA	15	20	If not zero jump to 2015
2025	MVI A,01	3E	01		Accumulator = 01
2027	STA 27F6	32	F6	27	Store value to 27F6
202A	JMP 2032	C3	32	20	Jump to 2032
202D	MVI A,0E	3E	0E		Set accumulator with 0E
202F	STA 27F6	32	F6	27	Store value to 27F6
2032	CALL 06FA	CD	FA	06	Call subroutine
2035	HLT	76			Stop

43. Write an assembly language program to find the given 8-bit number is Odd or Even using 8085 processor.

Address	HEX Codes	Label	Mnemonics	Comments
F000	3A, 00, 80		LDA 8000H	Load the number from memory
F003	E6, 01		ANI 01H	AND 01H with Acc content
F005	CA, 0D, F0		JZ EVEN	If Z = 0, it is Even
F008	3E, 01		MVI A, 01H	Load 01H to indicate it is Odd
F00A	C3, 0F, F0		JMP STORE	Jump to store
F00D	3E, FF	EVEN	MVI A, FFH	Load FFH to indicate it is Even
F00F	32, 50, 80	STORE	STA 8050H	Store the result into memory
F012	76		HLT	Terminate the program

44.Design of 3 stage pipeline for OR of two numbers using any high level language.

program:

```

counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1
b=int(input("ENTER NUMBER-2-"))
counter=counter+1
res= a or b
counter=counter+1
print(res)
counter=counter+2
INS=int(input("enter no. of instructions:"))
performance_measure=INS/counter
print("performance measure:",performance_measure)

```

OUTPUT:

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. A status bar at the bottom right indicates "In: 10 Col: 0". The code window displays the following session:

```
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\OR.py =====
ENTER NUMBER-1-1001
ENTER NUMBER-2-1010
1010
enter no. of instructions:4
performance measure: 0.8
>>>
```

45.Design of 3 stage pipeline for XOR of two numbers using any high level language.

```
counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1

b=int(input("ENTER NUMBER-2-"))
counter=counter+1

res= a ^ b
counter=counter+1

print(res)
counter=counter+2

INS=int(input("enter no. of instructions:"))

performance_measure=INS/counter

print("performance measure:",performance_measure)
```

OUTPUT:

```
idle shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\OR.py =====
ENTER NUMBER-1-1001
ENTER NUMBER-2-1010
1010
enter no. of instructions:4
performance measure: 0.8
>>>
```

46.Design of 2 stage pipeline for subtraction of two numbers using any high level language.

program:

```
counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1
b=int(input("ENTER NUMBER-2-"))
counter=counter+1
print("SUBTRACTION")

print("Enter Your Choice")
choice=int(input())
counter=counter+1
if choice==1:
    print("Performing Subtraction...")
    res=a-b
    counter=counter+1
if choice>=5:
    print("Enter Correct Input")
```

```

print(res)
counter=counter+1
print("CYCLE VALUE IS:",counter)
ins=int(input("Enter the No.of instructions:"))
performance_measure =ins/counter
print("performance measure is:" performance_measure)

```

OUTPUT:

```

IDLE Shell 3.10.2
File Edit Shell Debug Options Window Help
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\CA.py =====
ENTER NUMBER-1-23
ENTER NUMBER-2-18
1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION
Enter Your Choice
1
Performing Addition...
41
CYCLE VALUE IS: 4
Enter the No.of instructions:4
performance measure is: 1.0
>>>

```

49.Design of 4 stage pipeline for Division of two numbers using any high level language.

```

counter=1

a=int(input("ENTER NUMBER-1-"))
counter=counter+1

b=int(input("ENTER NUMBER-2-"))
counter=counter+1
print(" DIVISION")

print("Enter Your Choice")
choice=int(input())
if choice==1:

    print("Performing Addition...")
    res=a+b
    counter=counter+1

```

```
if choice==2:  
    print("Performing Subtraction...")  
    res=a-b  
    counter=counter+1  
  
if choice==3:  
    print("Performing Multiplication")  
    res=a*b  
    counter=counter+1  
  
if choice==4:  
    if b==0:  
        print("Denominator can't be Zero")  
    print("Performing Division")  
    res=a/b  
    counter=counter+1  
  
if choice>=5:  
    print("Enter Correct Input")  
  
print(res)  
counter=counter+3  
print("CYCLE VALUE IS:",counter)  
ins=int(input("Enter the No.of instructions:"))  
performance_measure =ins/counter  
print("performance measure is:",performance_measure)
```

OUTPUT:

The screenshot shows a window titled "IDLE Shell 3.10.2". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. Below the menu is a status bar with "Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32" and "Type "help", "copyright", "credits" or "license()" for more information." The main area displays the following text:

```
>>> ====== RESTART: C:\Users\vamsi\OneDrive\Desktop\py\CA.py ======
ENTER NUMBER-1-3
ENTER NUMBER-2-2
1-ADDITION 2-SUBTRACTION 3-MULTIPLICATION 4-DIVISION
Enter Your Choice
2
Performing Subtraction...
11
CYCLE VALUE IS: 7
Enter the No.of instructions:4
performance measure is: 0.5714285714285714
>>>
```

In the bottom right corner of the main window, there is a small icon of a person's head with a question mark inside. The status bar at the bottom right shows "Ln: 15 Col: 0".

