**NAME: MUDIREDDY VENKATA HARI OBULA REDDY**

**Reg no : 192011464**

**Subject name : CSA1288 (COMPUTER ARCHITECTURE )**

EXP

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

8 BIT ADDITION

LDA 2050

MOV B,A

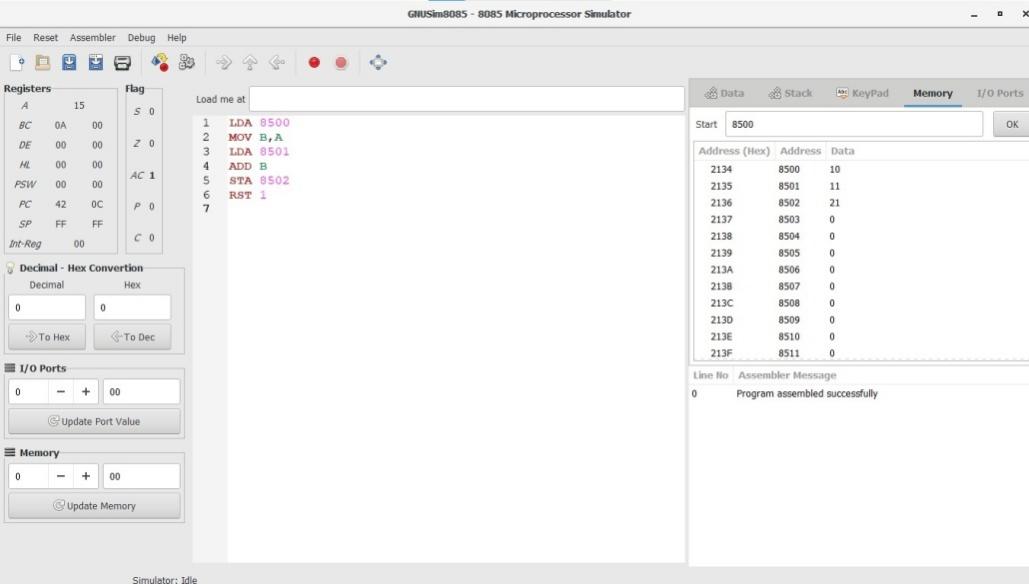
LDA 2052

ADD B

STA 2054

HLT

OUTPUT :



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 2

8 BIT SUBTRACTION

LDA 2050

MOV B,A

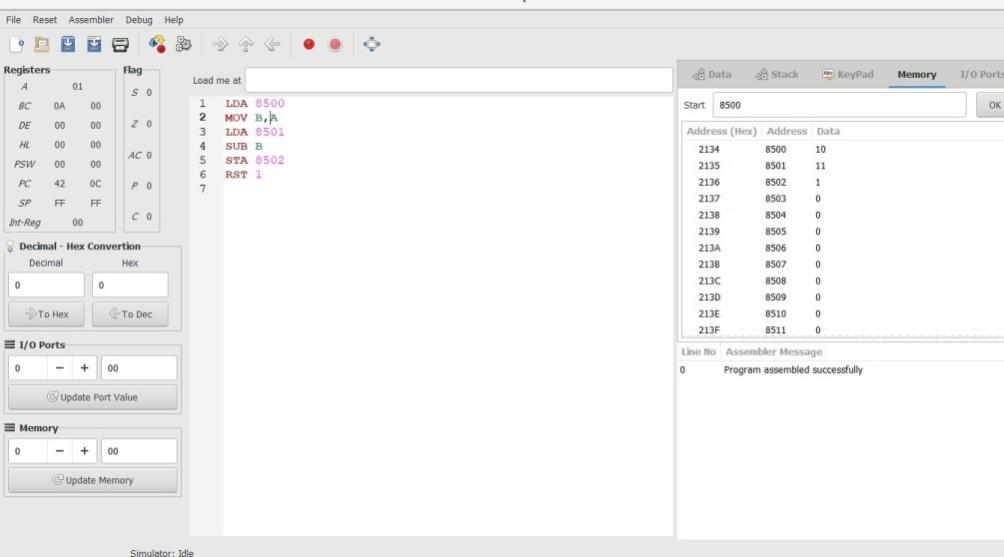
LDA 2052

SUB B

STA 2054

HLT

OUT PUT :



3. Write an assembly language program for adding two 16-bit datausing 8086 processor

16 BIT ADDITION

LHLD 2500

XCHG

LHLD 2502

MOV A,E

ADD L

MOV L,A

MOV A,D

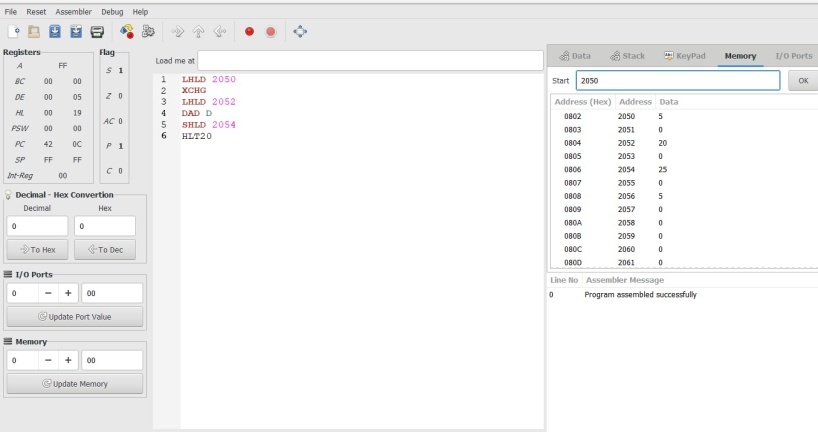
ADC H

MOV H,A

SHLD 2504

HLT

OUTPUT :



4. Write an assembly language program for subtracting two 16-bit datausing 8086 processor.

16 BIT SUBTRACTION

LHLD 2500

XCHG

LHLD 2502

MOV A,E

SUB L

MOV L,A

MOV A,D

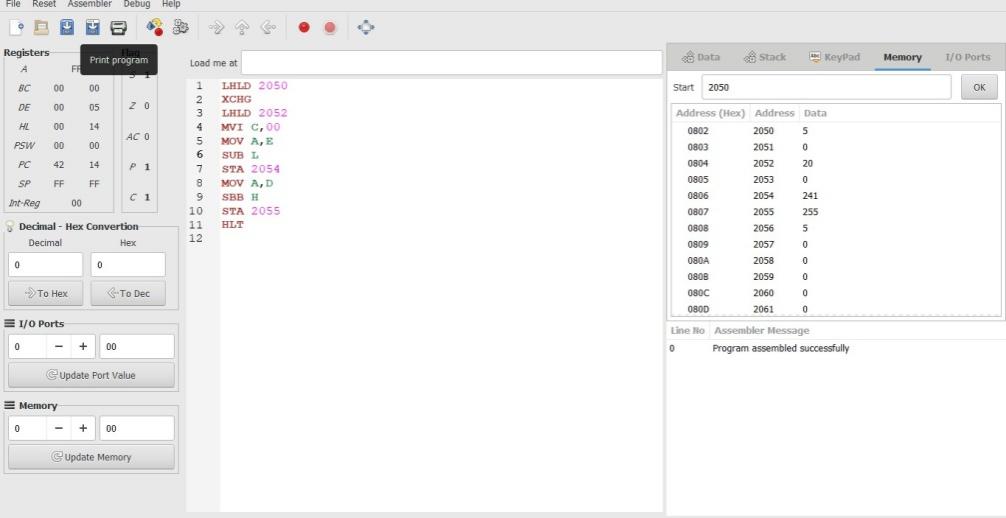
ADC H

MOV H,A

SHLD 2504

HLT

OUTPUT :



5. **8 BIT MULTIPLICATION**

MVI D,00

MVI A,00

LXI H,4150

MOV B.,M

INX H

MOV C,M

LOOP: ADD B

JNC NEXT

INR D

NEXT: DCR C

JNZ LOOP

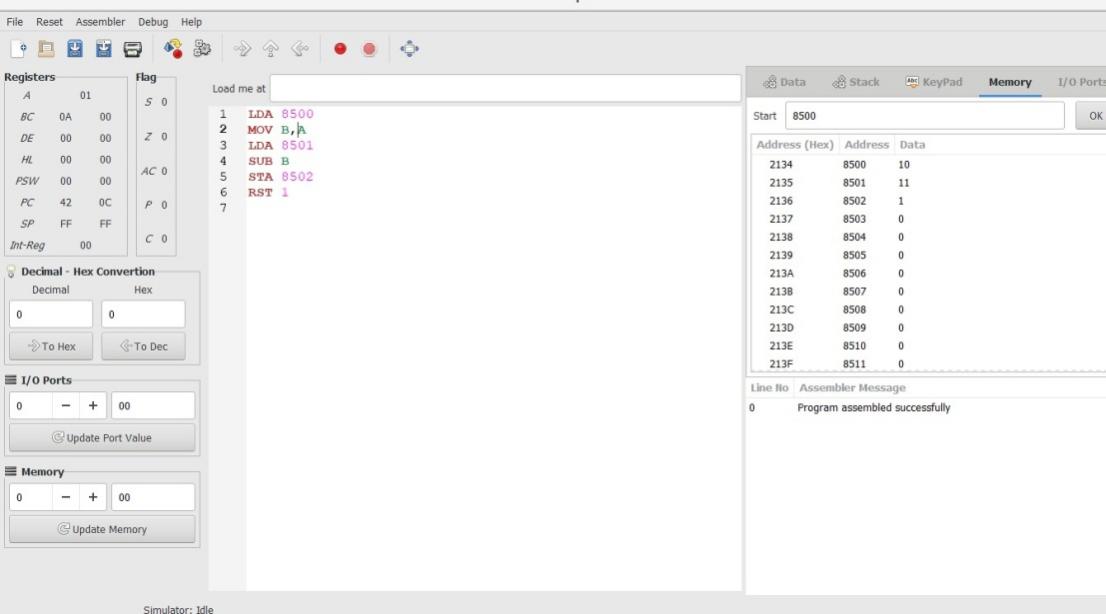
STA 4152

MOV A,D

STA 4153

HLT

OUTPUT :



6. 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

8 BIT DIVISION

LXI H,1100

MOV B,M

MVI C,00

INX H

MOV A,M

LOOP: CMP B

JC SKIP

SUB B

INR C

JMP LOOP

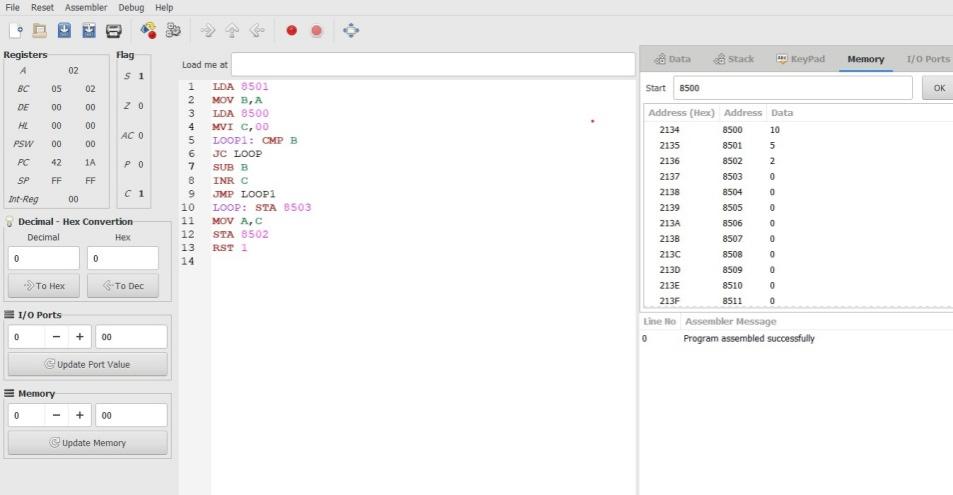
SKIP: STA 1102

MOV A,C

STA 1103

HLT

OUTPUT :



**7. Write an assembly language program for multiplying two 16-bit datausing 8086 processor.**

16 BIT MULTIPLICATION

LHLD 2050

SPHL

LHLD 2052

XCHG

LXI H,0000H

LXI B,0000H

DAD SP

JNC 2013

INX B

DCX D

MOV A,E

ORA D

JNZ 200E

SHLD 2054

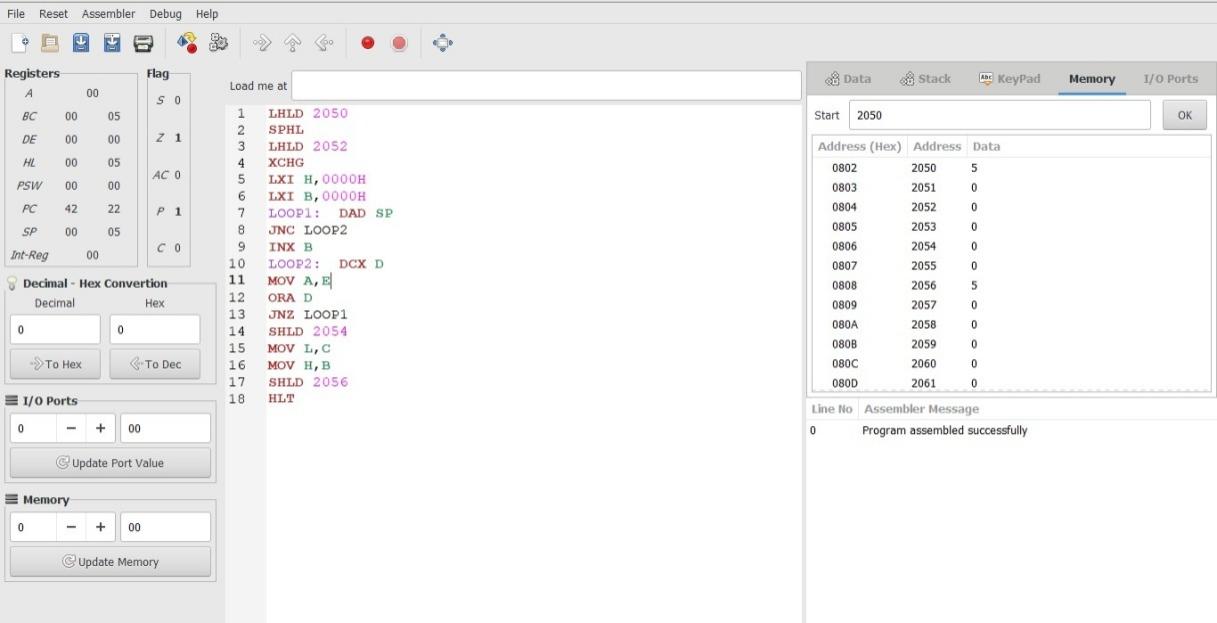
MOV L,C

MOV H,B

SHLD 2056

HLT

OUTPUT :



**8 Write an assembly language program for dividing two 16-bit datausing 8086 processor.**

16 BIT DIVISION

LXI B,0000H

LHLD FC02H

XCHG

LHLD FC00H

MOV A,L

SUB E

MOV L,A

MOV A,H

SBB D

MOV H,A

JC SKIP

INX B

JMP LOOP

DAD D

SHLD FC06H

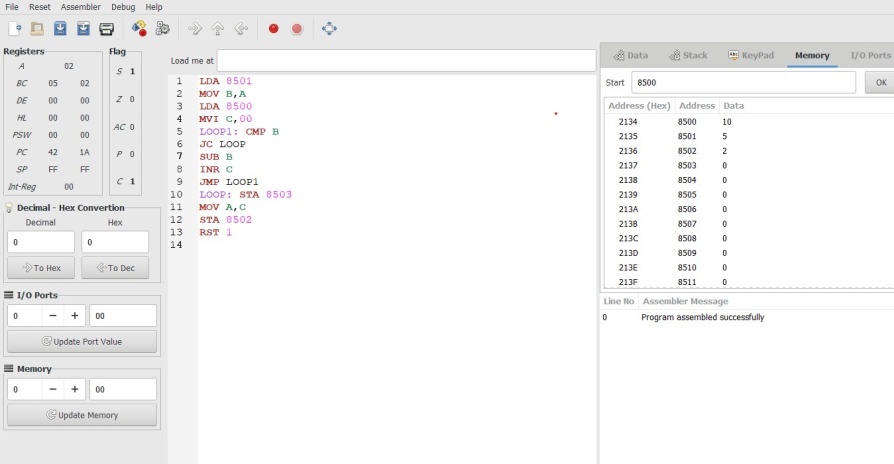
MOV L,C

MOV H,B

SHLD FC04H

HLT

**OUTPUT :**

****

**EXP : 09**

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

LDA 2050

MOV B, A

LDA 2051

CMP B

JNC 2053

MOV A, B

STA 2060

HLT

**EXP :10**

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

LHLD 2050

XCHG

LHLD 2052

MVI C,00

MOV A,E

SUB L

STA 2054

MOV A,D

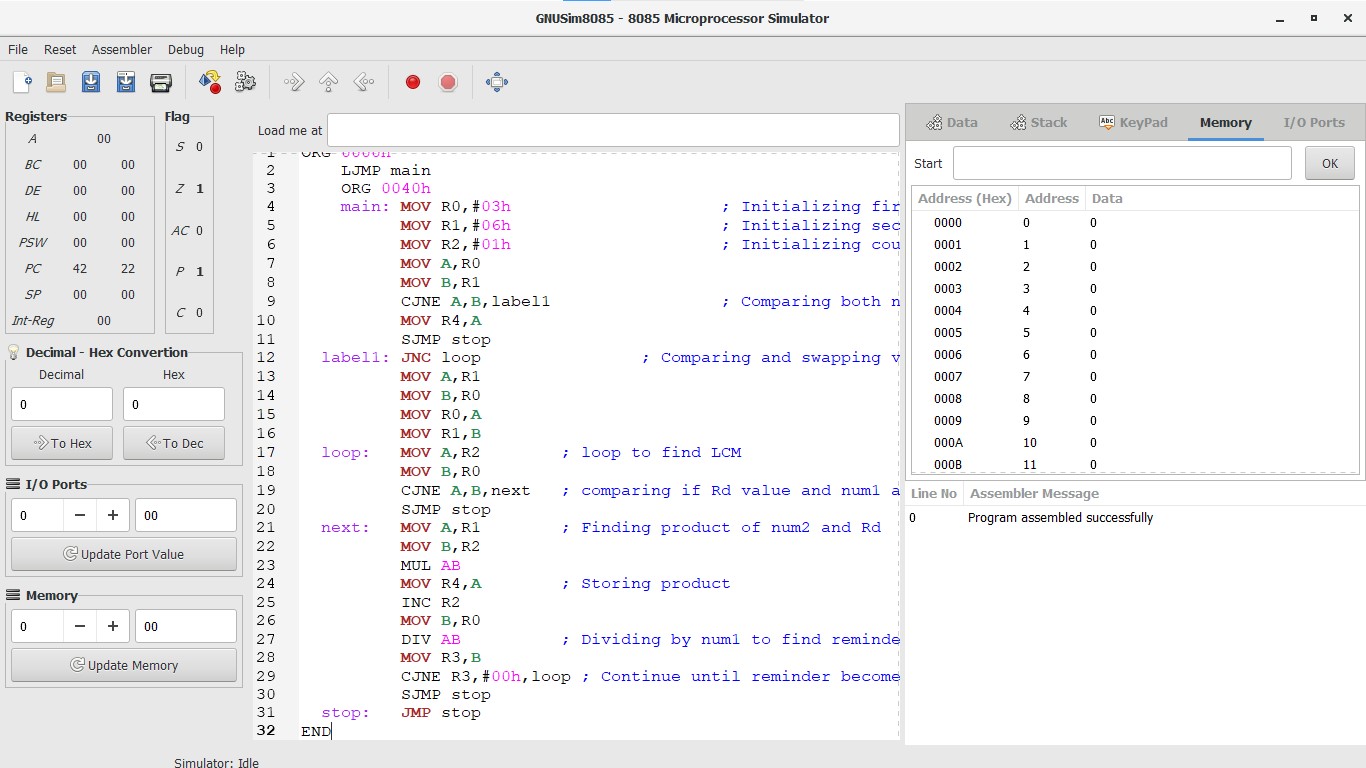
SBB H

STA 2055

HLT

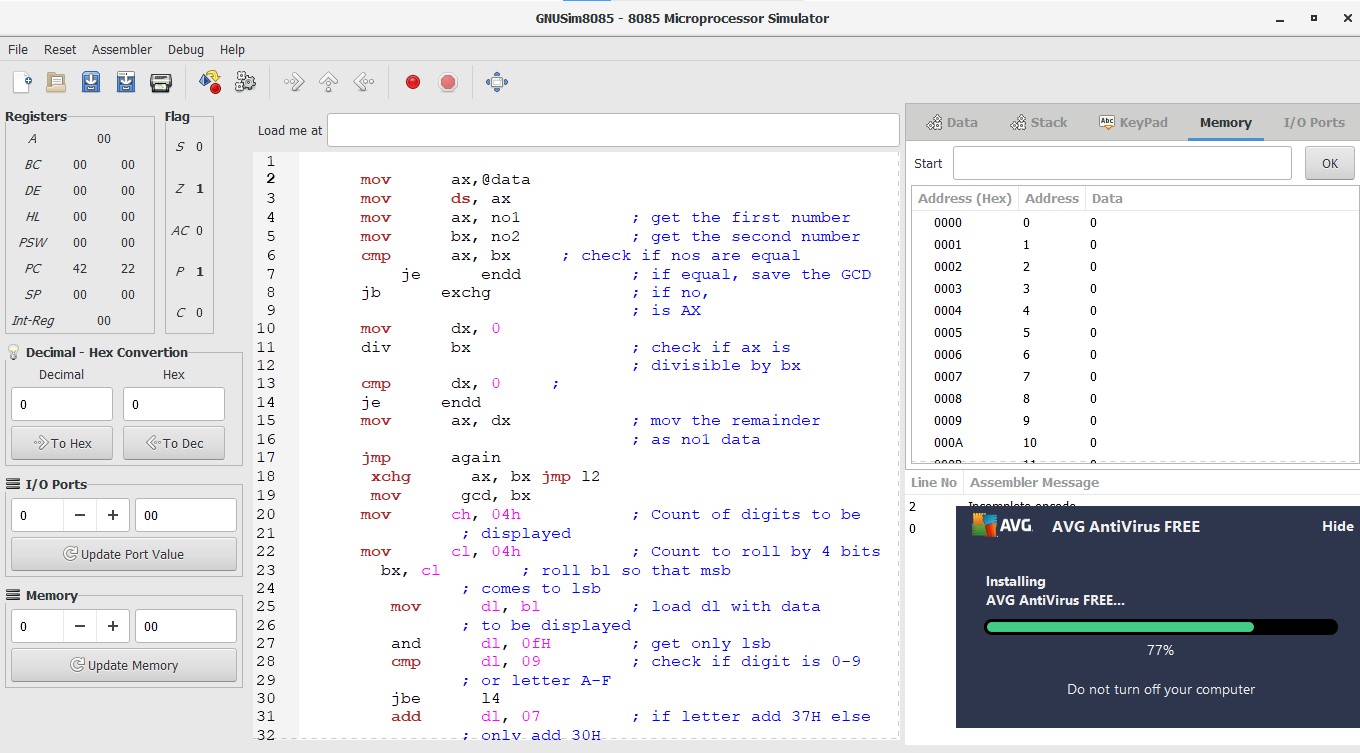
**EXP :11**

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

****

**EX :12**

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

****

**EX :13**

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

**EXP :14**

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

def hex:

num=int(input("enter the number:"))

print("of the number is")

print(hex(n))

**EXP :15**

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

def convert(num):

octalDigit = 0

count = 1

i = 0

pos = 0

octalArray = [0] \* 32

while num != 0:

digit = num % 10

octalDigit += digit \* pow(2, i)

i += 1

num //= 10

octalArray[pos] = octalDigit

if count % 3 == 0:

octalDigit = 0

i = 0

pos += 1

count += 1

for j in range(pos, -1, -1):

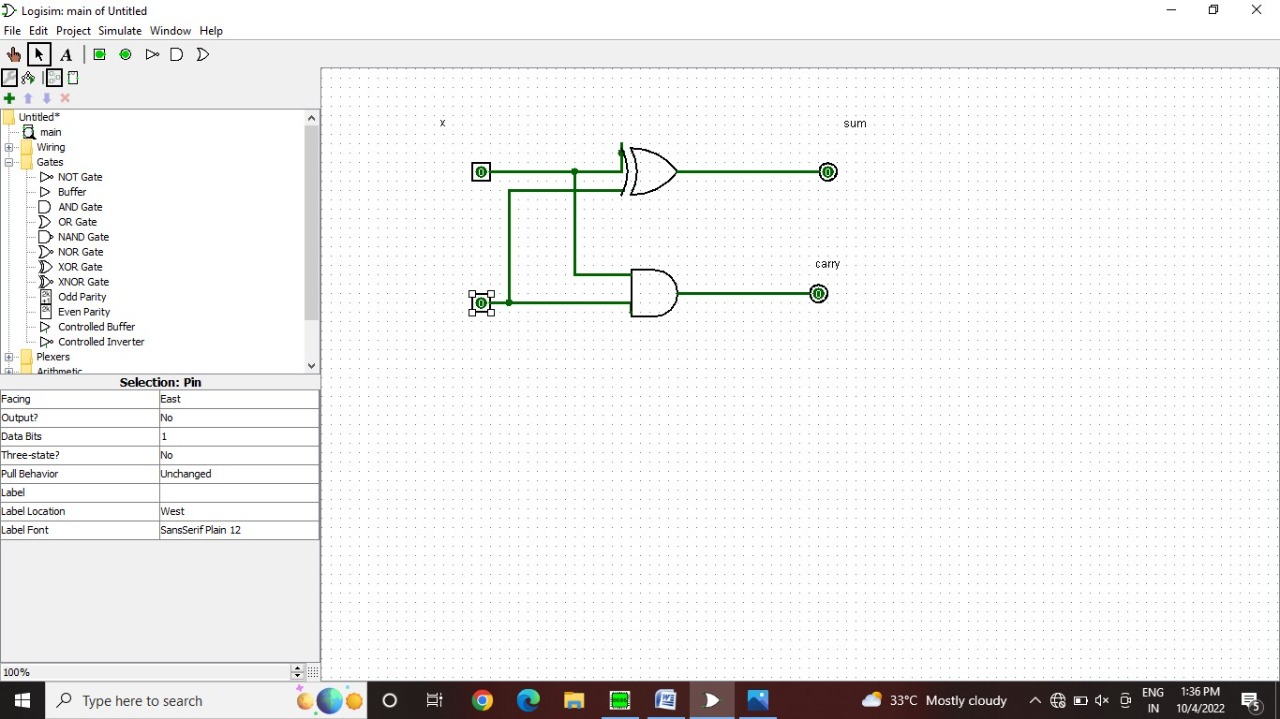
print(octalArray[j], end='')

binary = 1010

convert(binary)

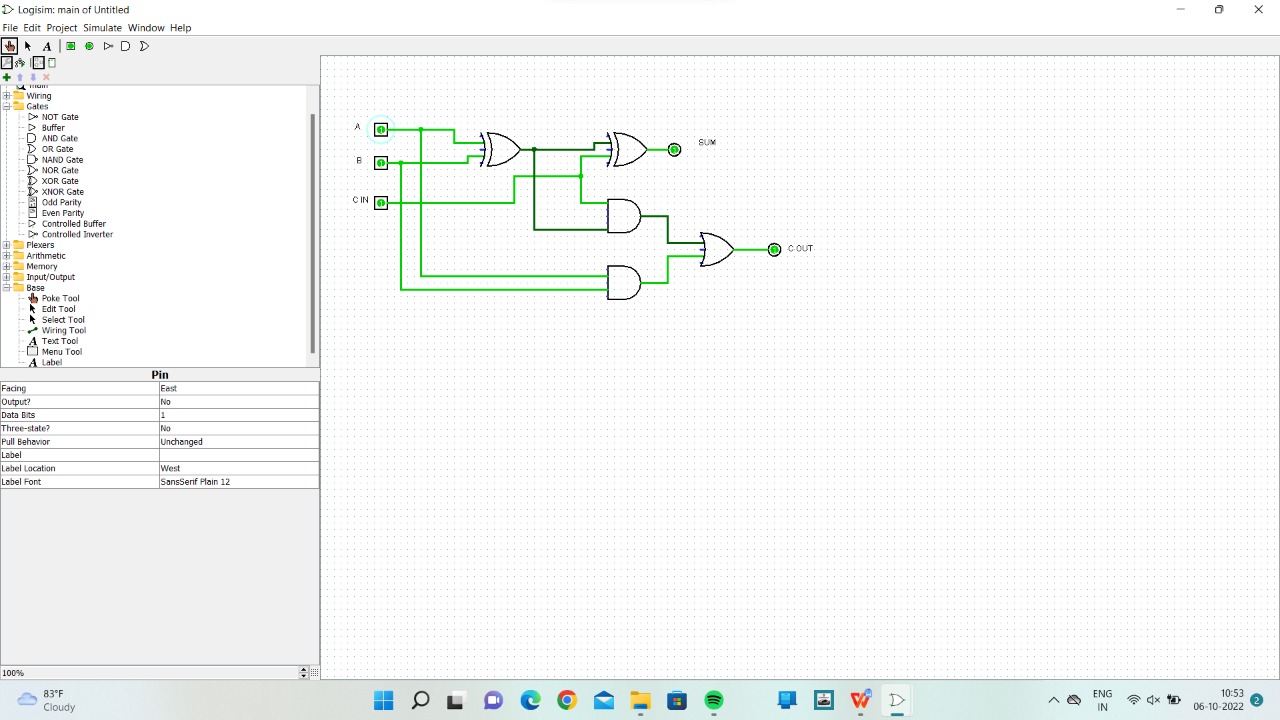
16. Design and implement 2-bit half adder using logisim simulator

AIM :To design and implement the two bit half adder using Logisim simulator.

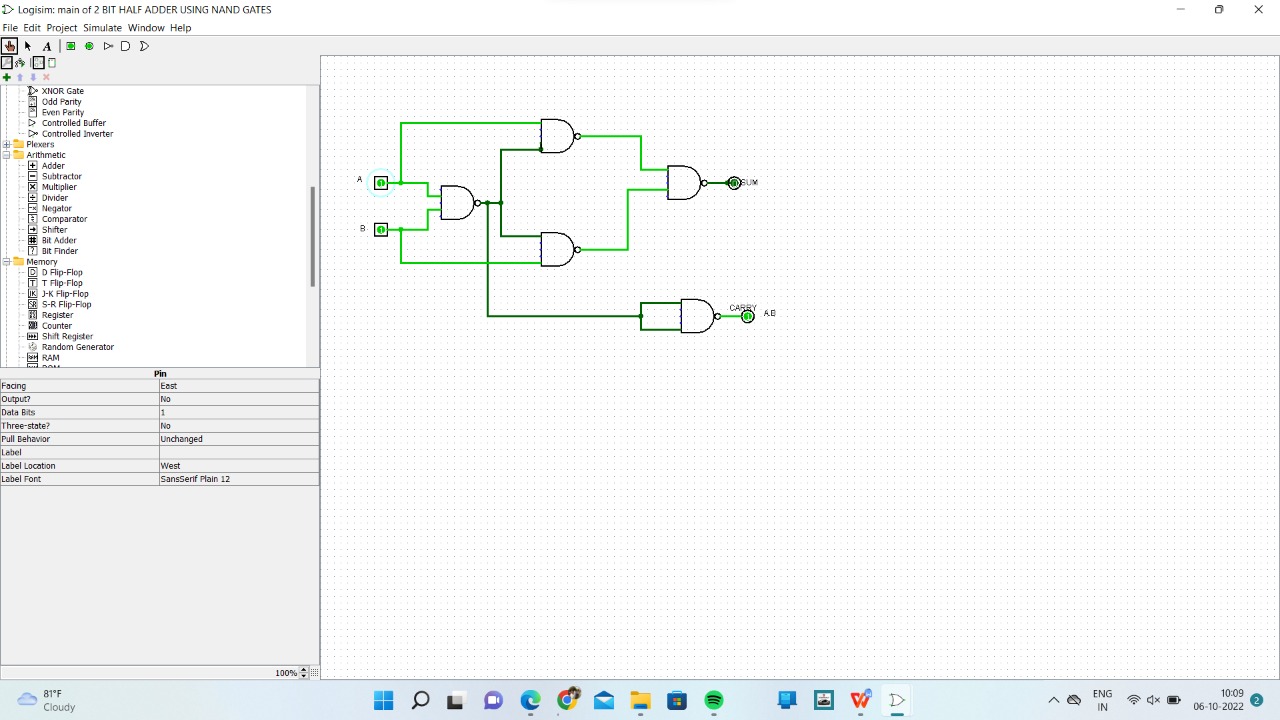


17. Design and implement 3-bit full adder using logisim simulator

THREE BIT FULL ADDER



18. Design and implement 2-bit half adder with NAND gatesusing logisim simulator.



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

def binToHexa(n):

bnum = int(n)

temp = 0

mul=1

count = 1

hexaDeciNum = ['0'] \* 100

i = 0

while bnum != 0:

rem = bnum % 10

temp = temp + (rem\*mul)

if count % 4 == 0

if temp < 10:

hexaDeciNum[i] = chr(temp+48)

else:

hexaDeciNum[i] = chr(temp+55)

mul = 1

temp = 0

count = 1

i = i+1

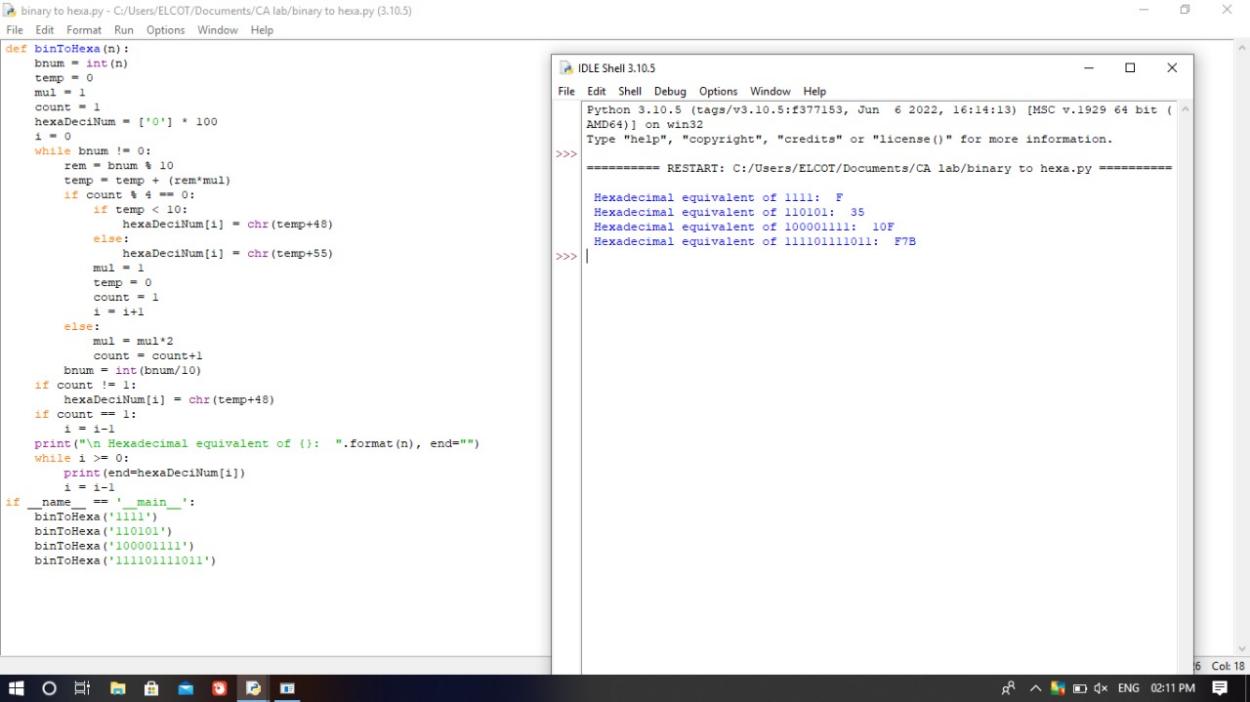
else:

mul = mul\*2

count = count+1

bnum = int(bnum/10)

output:



22. Design of 2 stage pipeline for addition 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("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:

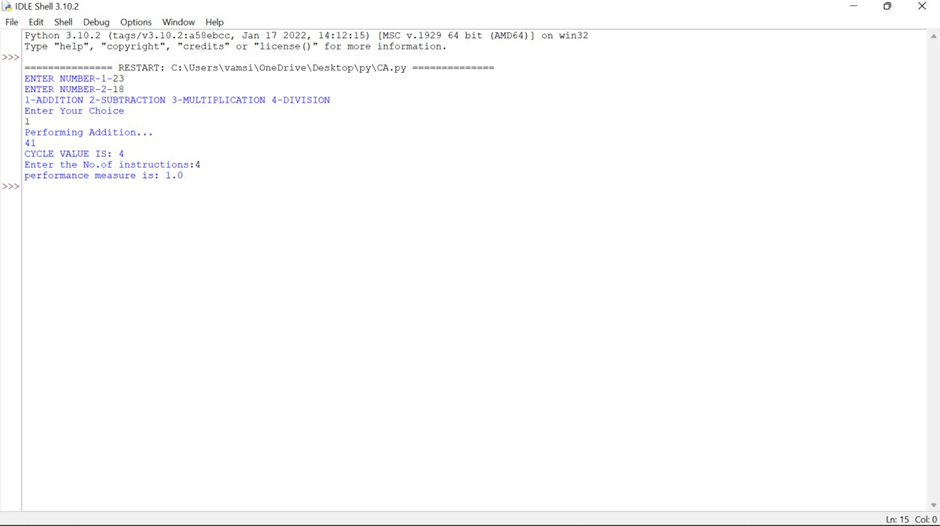
print("Enter Correct Input")

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 :



23: Design of 2 stage pipeline for subtraction 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("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:

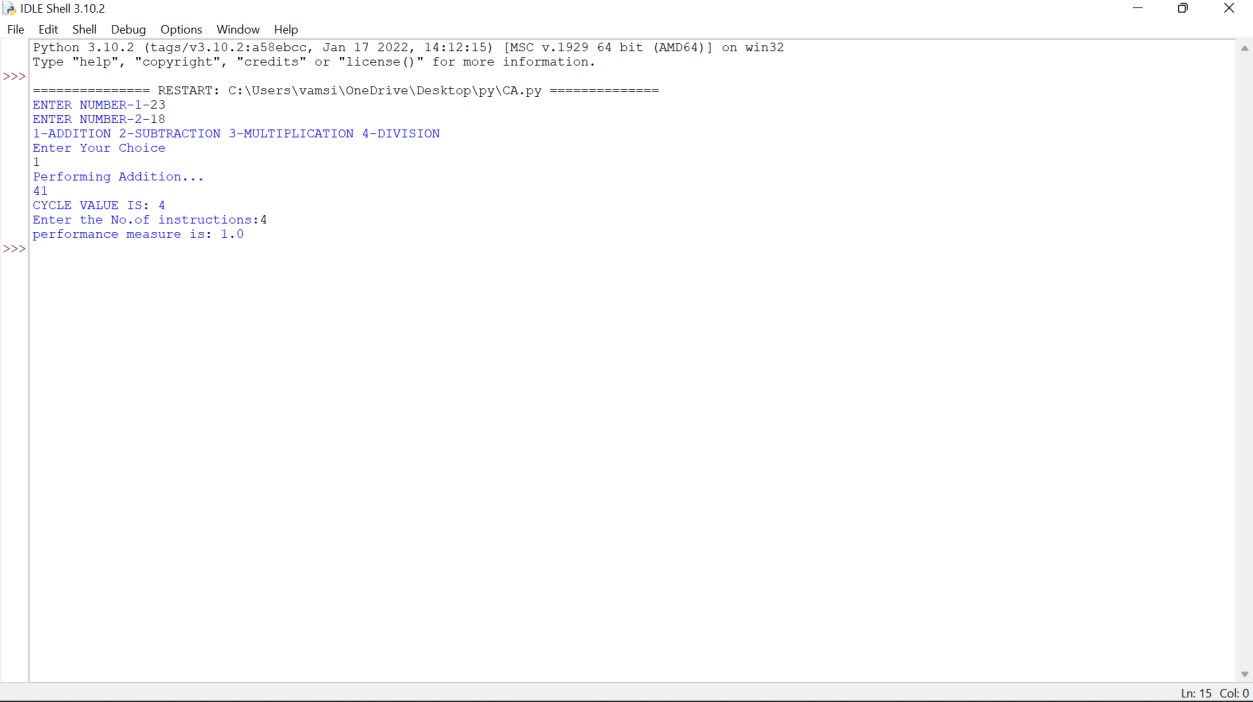
print("Enter Correct Input")

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:



**EXP:24**

**Design of 3stage pipeline for AND 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

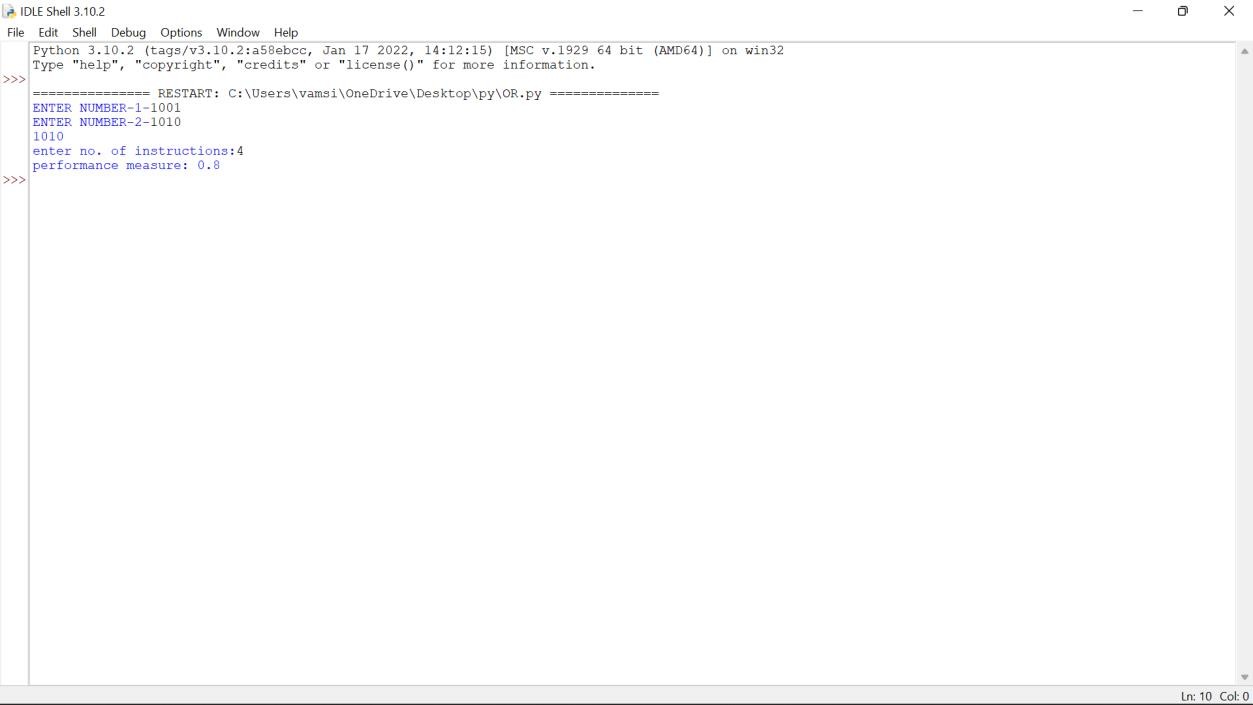
res= a and b counter=counter+1 print(res) counter=counter+2

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

performance\_measure=INS/counter

print("performance measure:",performance\_measure)

OUTPUT :



**EXP :25**

**Design of 3stage pipeline for OR 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

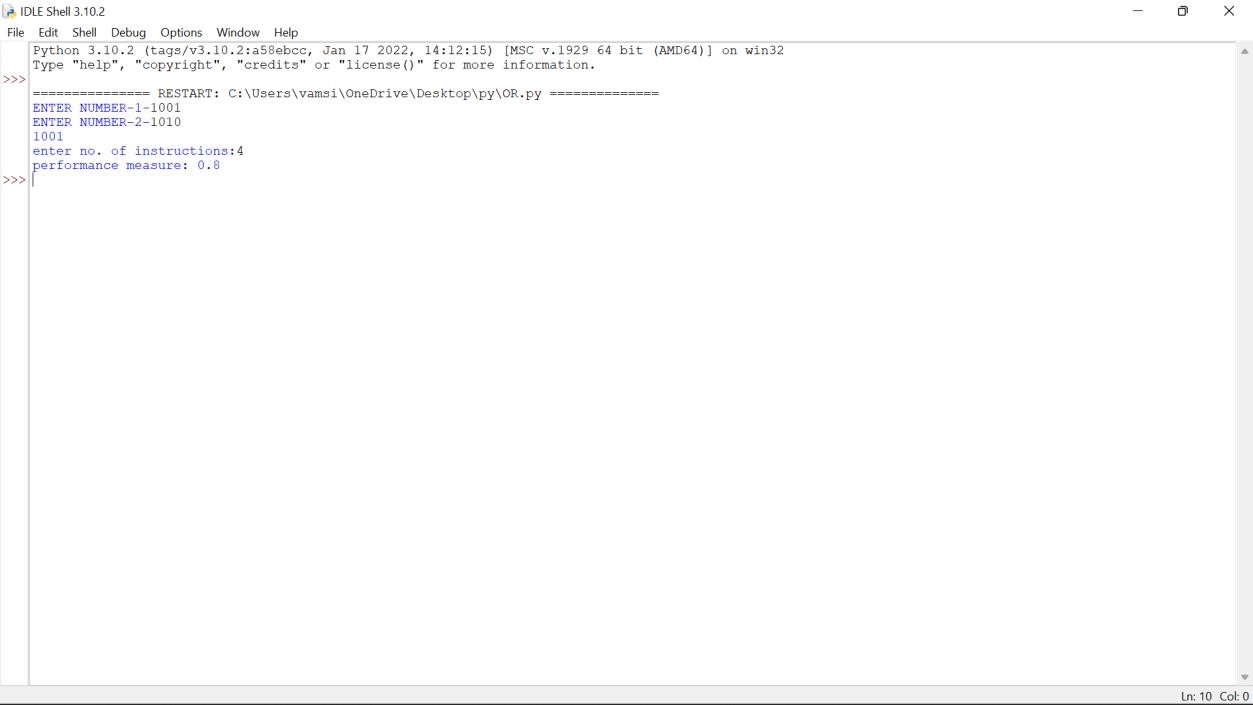
res= a or b counter=counter+1 print(res) counter=counter+2

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

performance\_measure=INS/counter

print("performance measure:",performance\_measure)

OUTPUT :



28. Design of 4 stage pipeline for Multiplication 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("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:

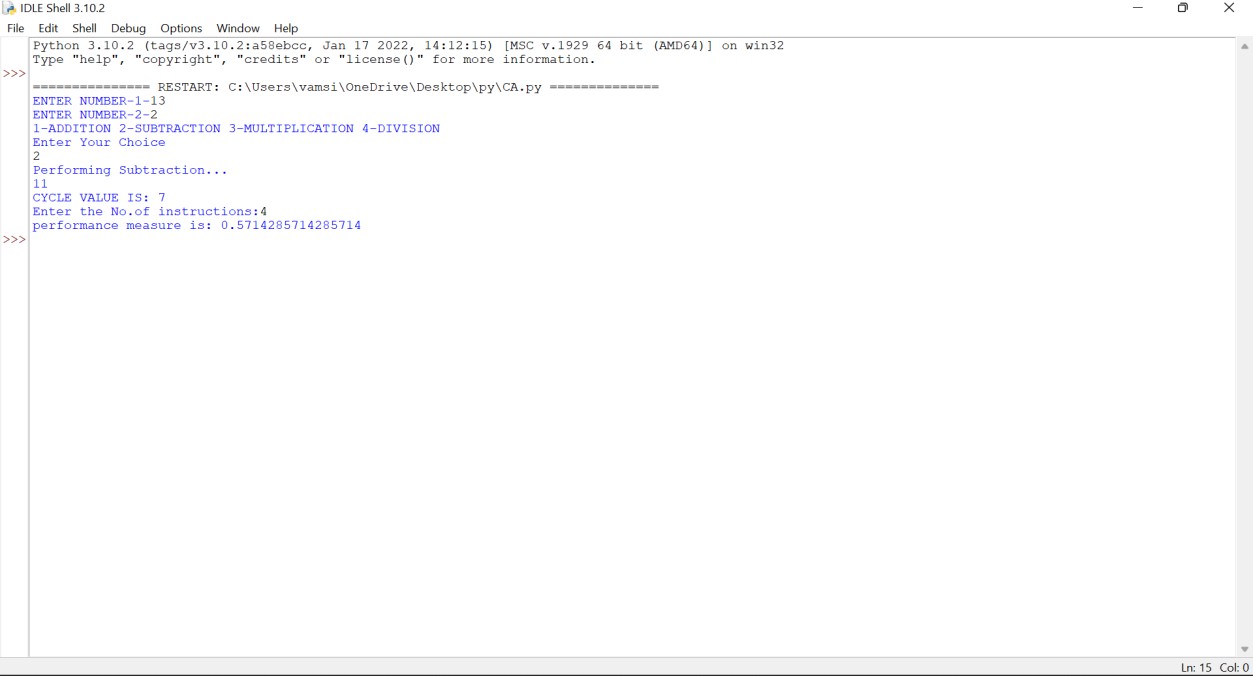
print("Enter Correct Input")

print(res) counter=counter+3

print("CYCLE VALUE IS:",counter) ins=int(input("Enter the No.of instructions:")) performance\_measure =ins/counter

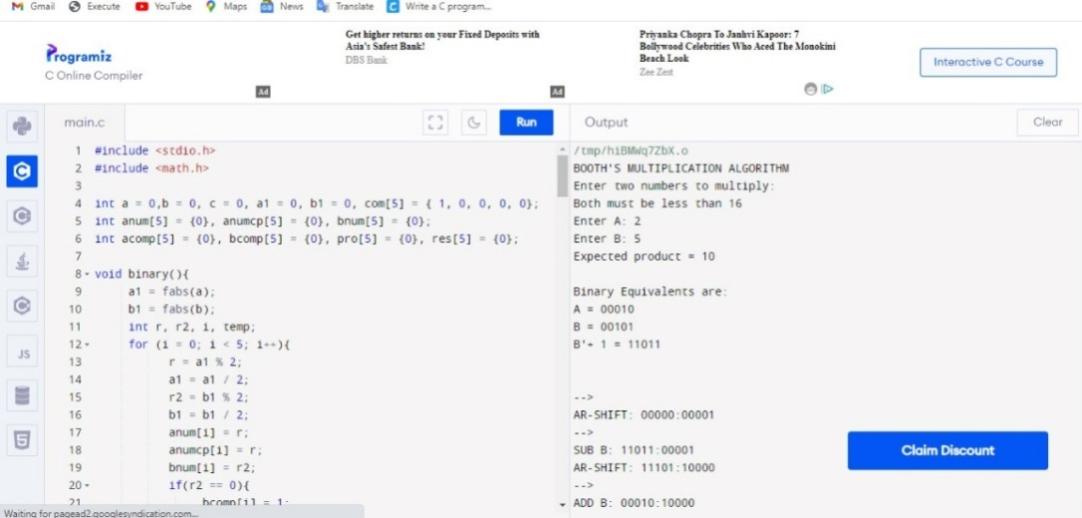
print("performance measure is:",performance\_measure)

OUTPUT :



**EXP :29**

**Write a program to perform Booth’s multiplication of two signed numbers using any high level language**



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

#include<stdlib.h>

#include<stdio.h>

int acum[100]={0} ;

void add(int acum[],int b[],int n);

int q[100],b[100];

int main()

{

int x,y;

printf("Enter the Number :");

scanf("%d%d",&x,&y);

int i=0;

while(x>0||y>0)

{

if(x>0)

{

q[i]=x%2;

x=x/2;

}

else

{

q[i]=0;

}

if(y>0)

{

b[i]=y%2;

y=y/2;

}

else

{

b[i]=0;

}

i++;

}

int n=i;

int bc[50];

printf("\n");

for(i=0;i<n;i++)

{

if(b[i]==0)

{

bc[i]=1;

}

else

{

bc[i]=0;

}

}

bc[n]=1;

for(i=0;i<=n;i++)

{

if(bc[i]==0)

{

bc[i]=1;

i=n+2;

}

else

{

bc[i]=0;

}

}

int l;

b[n]=0;

int k=n;

int n1=n+n-1;

int j,mi=n-1;

for(i=n;i!=0;i--)

{

for(j=n;j>0;j--)

{

acum[j]=acum[j-1];

}

acum[0]=q[n-1];

for(j=n-1;j>0;j--)

{

q[j]=q[j-1];

}

add(acum,bc,n+1);

if(acum[n]==1)

{

q[0]=0;

add(acum,b,n+1);

}

else

{

q[0]=1;

}

}

printf("\nQuoient : ");

for( l=n-1;l>=0;l--)

{

printf("%d",q[l]);

}

printf("\nRemainder : ");

for( l=n;l>=0;l--)

{

printf("%d",acum[l]);

}

return 0;

}

void add(int acum[],int bo[],int n)

{

int i=0,temp=0,sum=0;

for(i=0;i<n;i++)

{

sum=0;

sum=acum[i]+bo[i]+temp;

if(sum==0)

{

acum[i]=0;

temp=0;

}

else if (sum==2)

{

acum[i]=0;

temp=1;

}

else if(sum==1)

{

acum[i]=1;

temp=0;

}

else if(sum==3)

{

acum[i]=1;

temp=1;

}

}

}

OUTPUT :

Enter the Number :8

3

Quoient : 0010

Remainder : 00010

**EXP :31**

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

HIT RATIO AND MISS CACHE

#include <stdio.h>

int main() {

float cachehit, cachemiss;

float cachehitratio;

printf("\n enter the total number of cache hits:");

scanf("%d",&cachehit);

printf("\n enter the number of cache misses:");

scanf("%d",&cachemiss);

cachehitratio=cachehit/(cachehit+cachemiss);

printf("\n Cache Hit Ratio: %f",cachehitratio);

printf("\n Cache Miss Ratio: %f",1-cachehitratio);

return 0;

}

OUTPUT :

enter the total number of cache hits:43

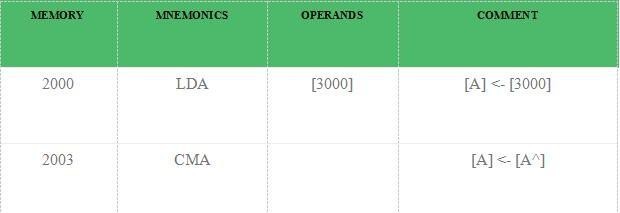
enter the number of cache misses:11

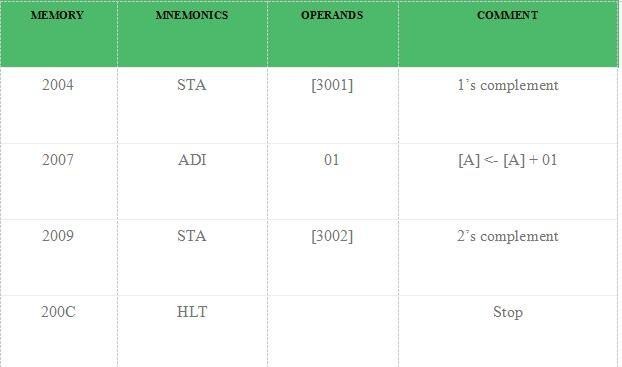
Cache Hit Ratio: 0.796296

Cache Miss Ratio: 0.20370

**EXP 32:** Write an assembly language program to find 1’s complement of 8 bit number.

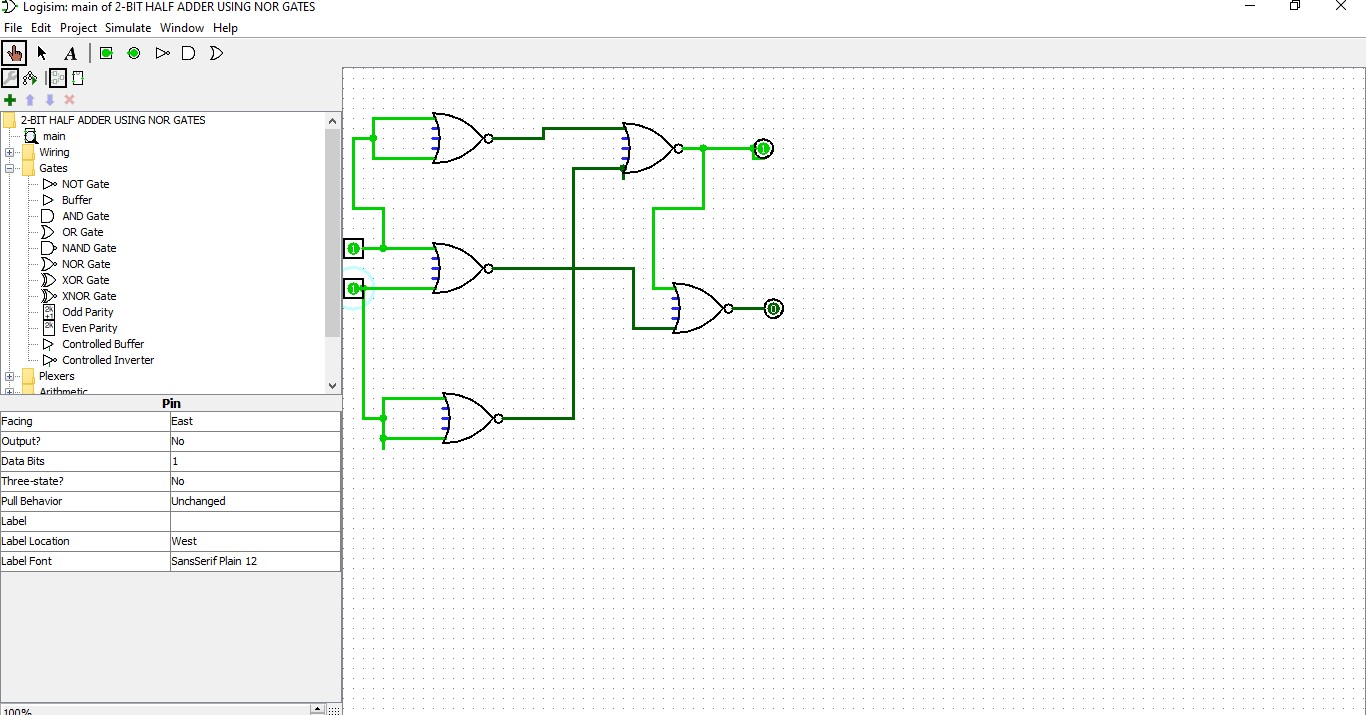
**EXP 33.** Write an assembly language program to find 2’s complement of 8 bit number.





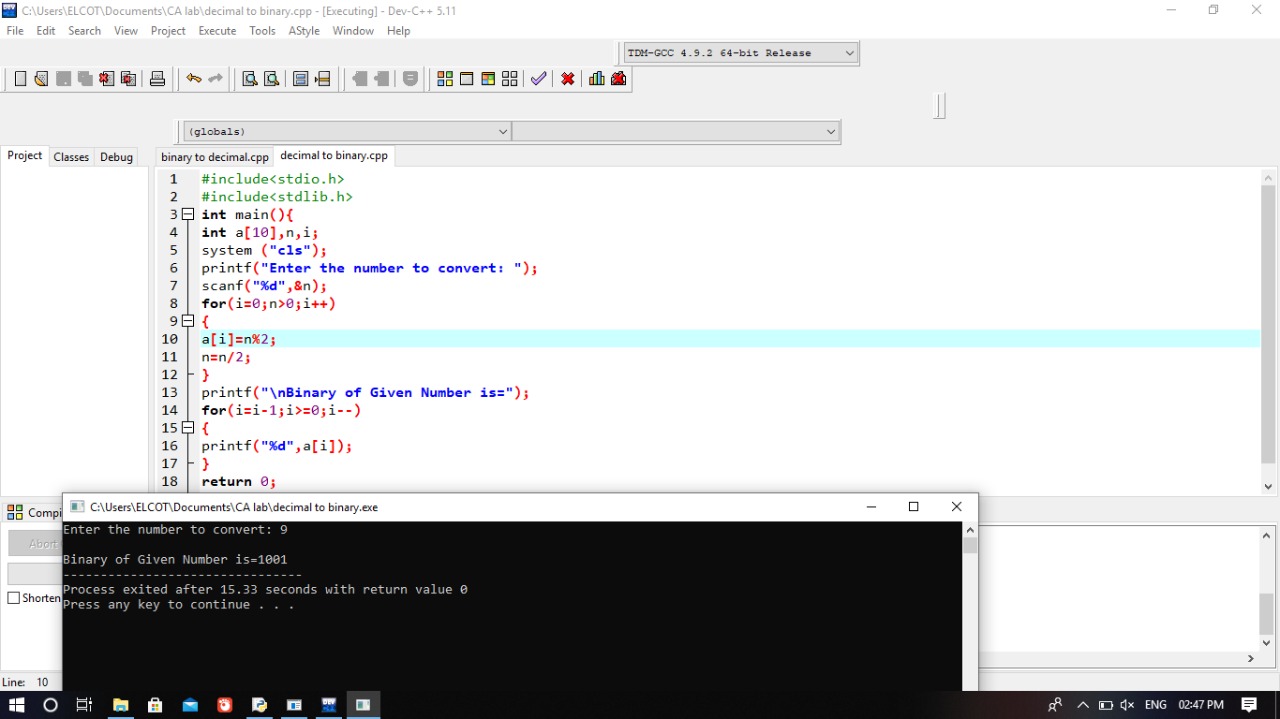
**EXP :35**

1. Design and implement 2-bit half adder with NOR gates using logisim simulator.

****

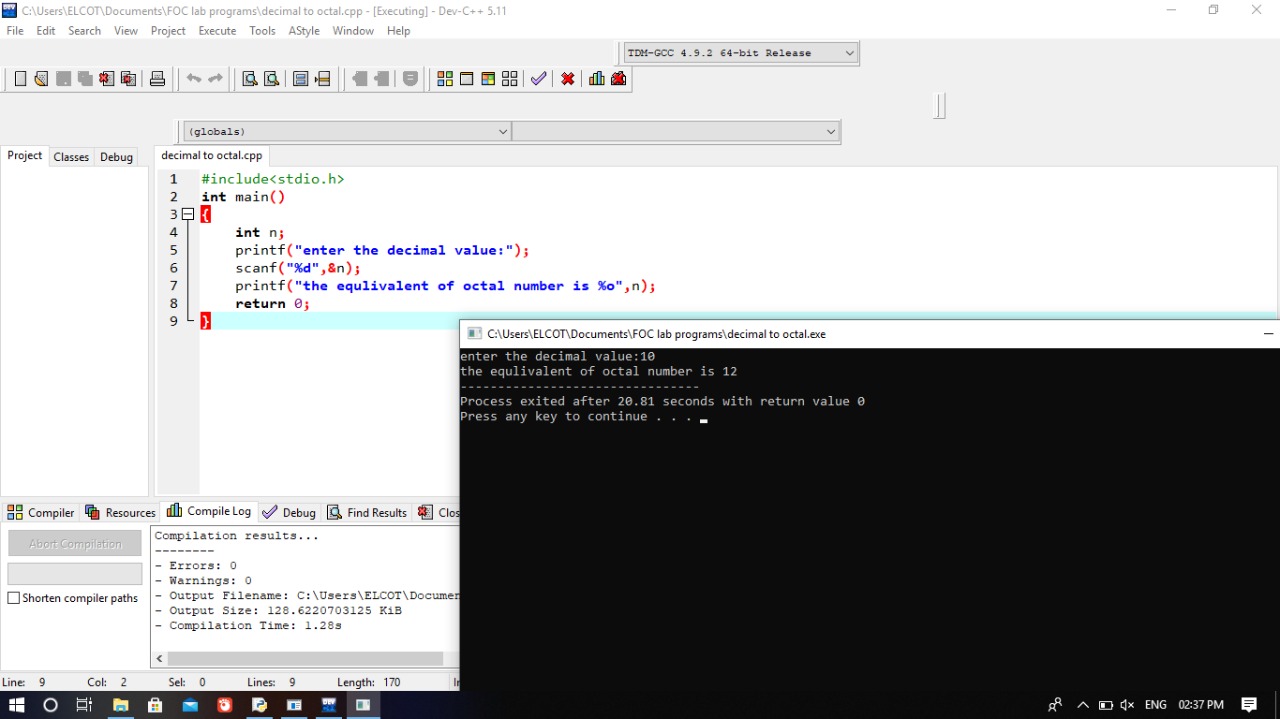
**EXP: 36**

**DECIMAL TO BINARY**

****

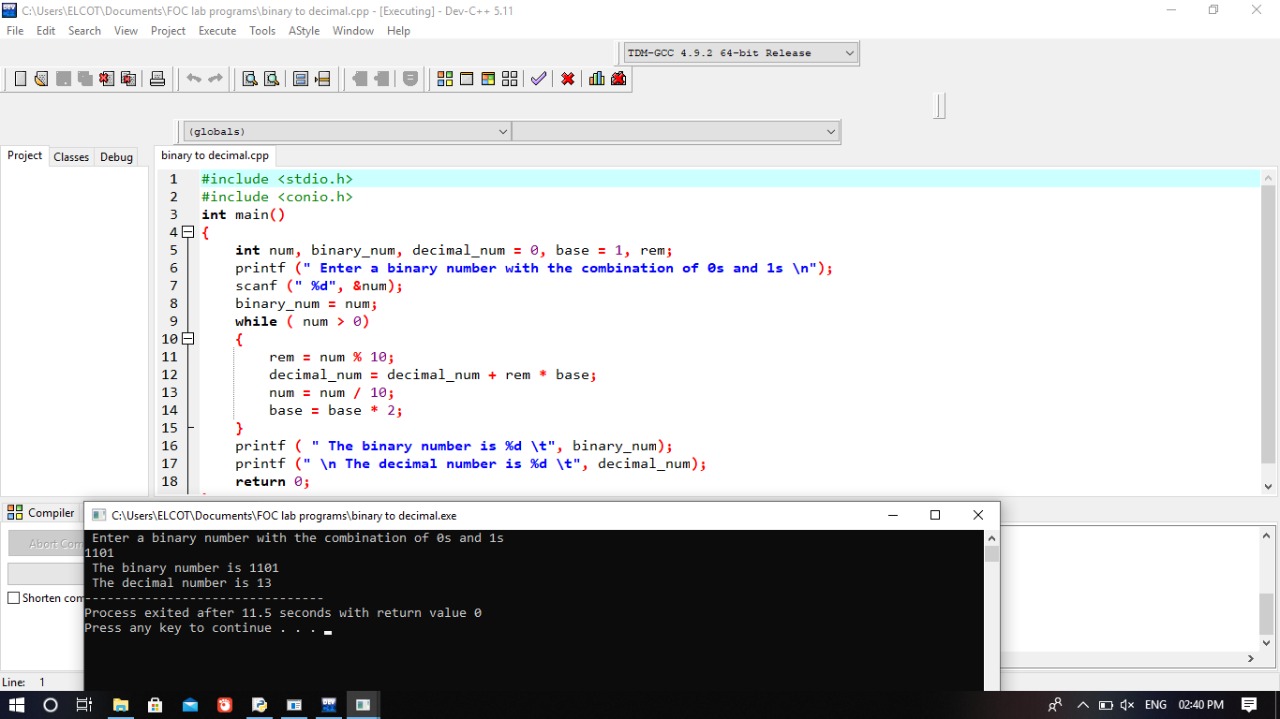
**EXP : 37**

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

****

**EXP 38**

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

****

**EXP :39**

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

#include <stdio.h>

int main()

{

float cr; int p,p1,i;

float cpu[5]; float cpi,ct,max; int n=1000; for(i=0;i<=4;i++)

{

cpu[5]=0;

}

printf("\n Enter the number of processors:"); scanf("%d",&p);

p1=p; for(i=0;i<p;i++)

{

printf("\n Enter the Cycles per Instrcution of processor:"); scanf("%f",&cpi);

printf("\n Enter the clockrate in GHz:"); scanf("%f",&cr);

ct=1000\*cpi/cr;

printf("The CPU time is: %f",ct); cpu[i]=ct;

}

max=cpu[0];

//printf("%f", max); for(i=0;i<p1;i++)

{

if(cpu[i]<=max) max=cpu[i];

}

printf("\n The processor has lowest Execution time is: %f ", max); return 0;

}

**OUTPUT :**

Enter the number of processors:3

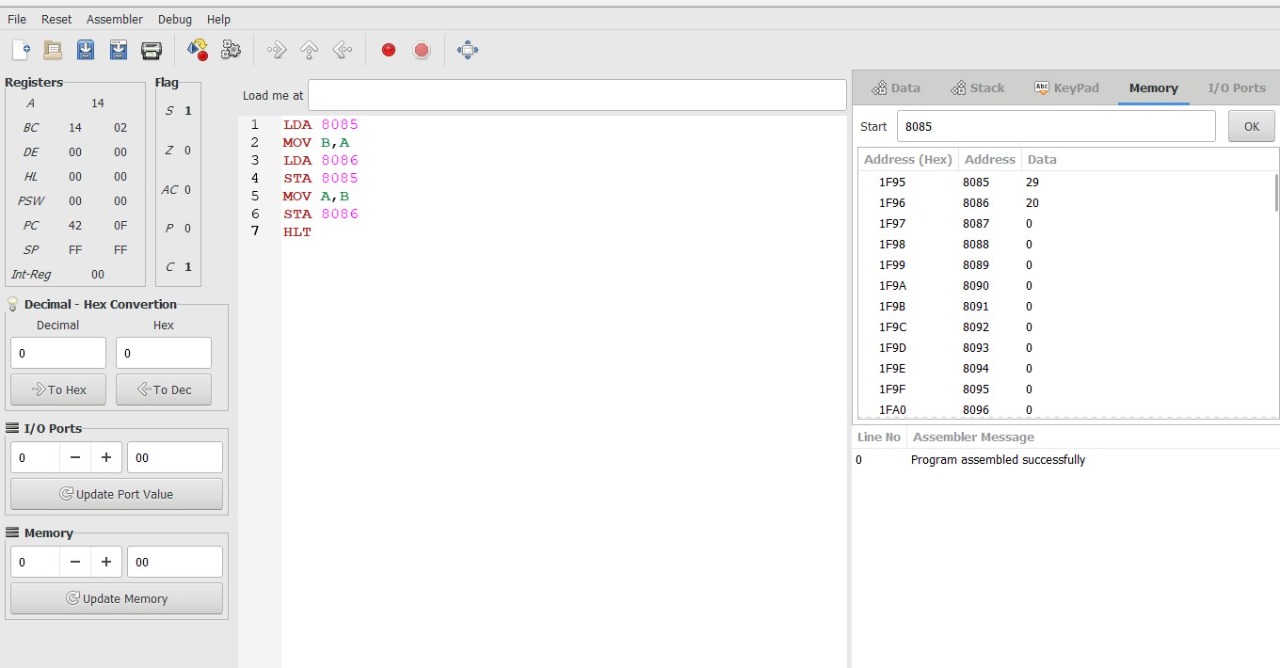
Enter the Cycles per Instrcution of

processor:1.5 Enter the clockrate in GHz:3

The CPU time is: 500.000000

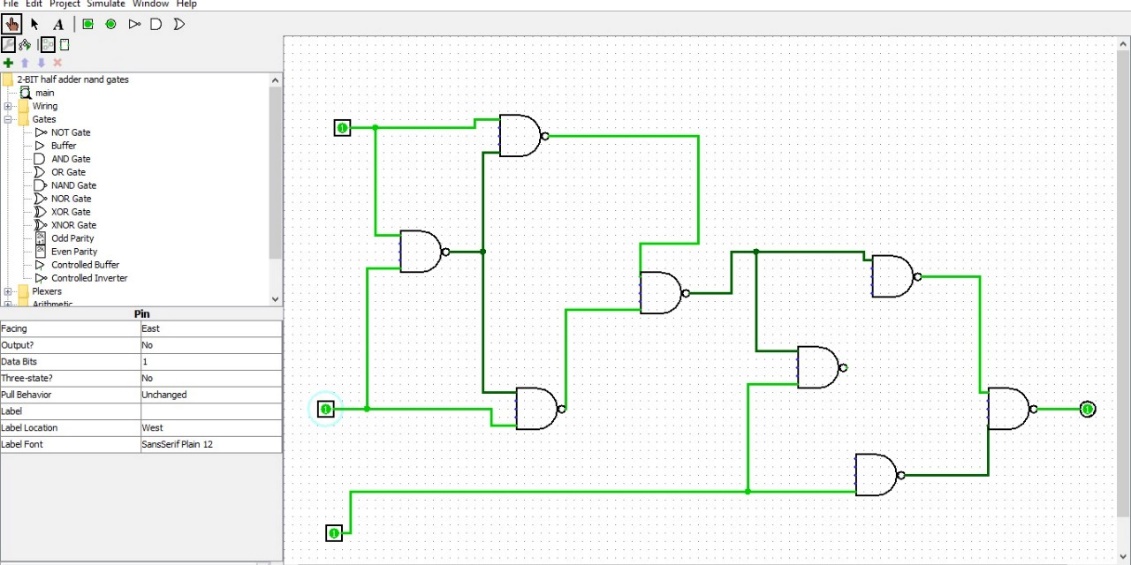
**EXP :40**

Write an assembly language program to swap two 8-bit datausing 8085 processor

****

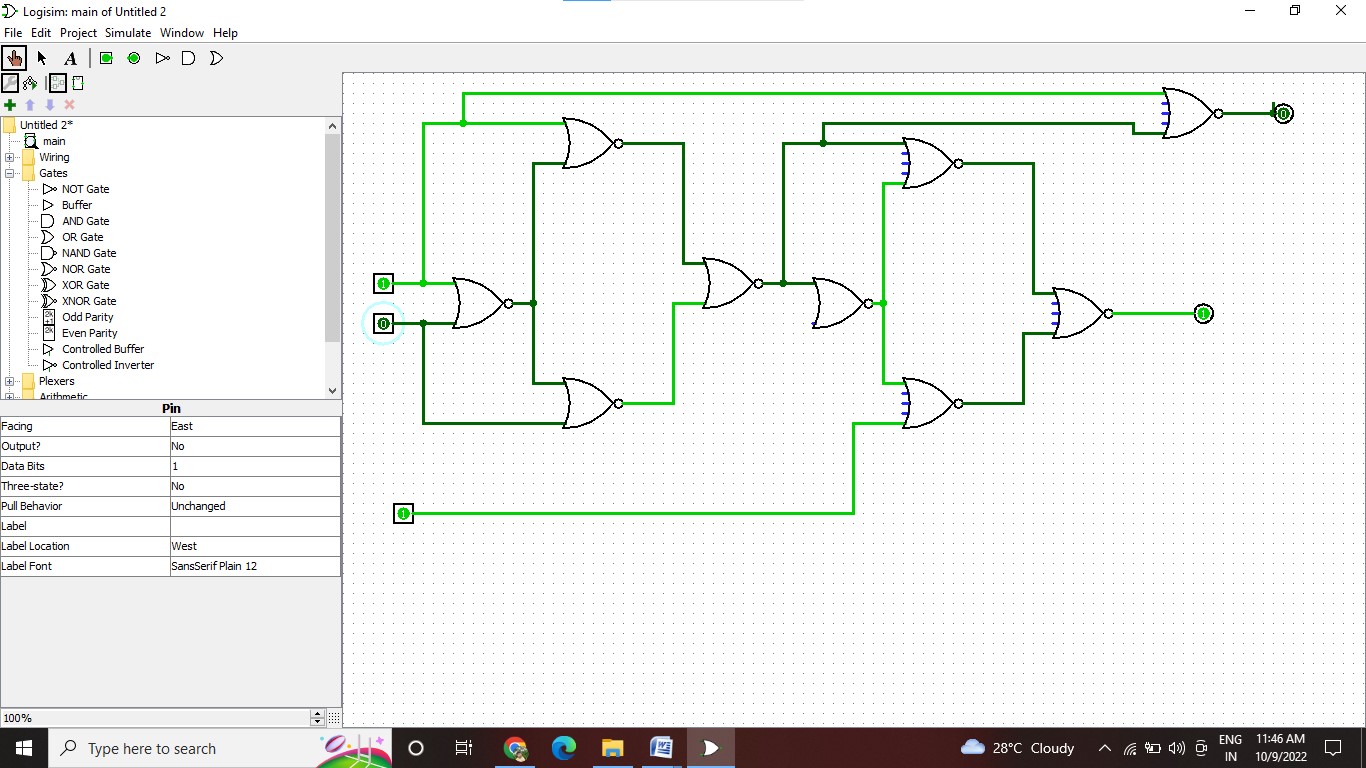
**EXP : 43**

Design and implement 3-bit Full adder with NAND gates using logisim simulator.

****

**EXP 44**

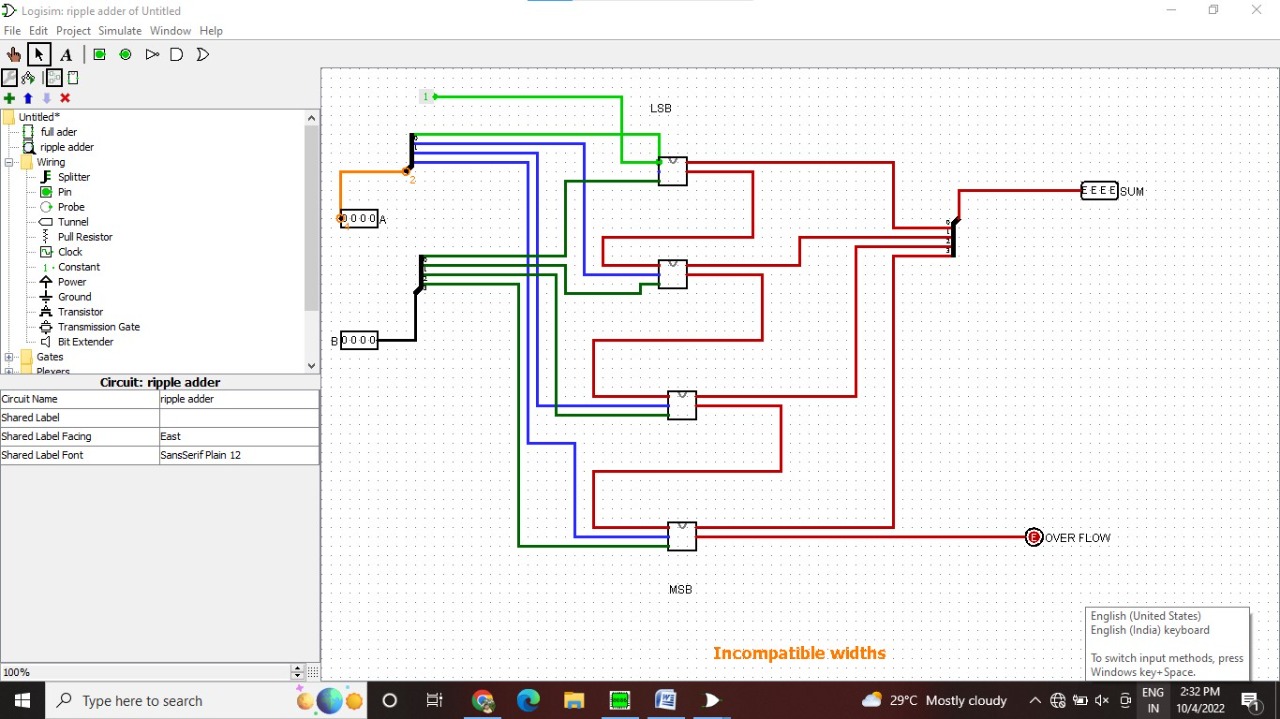
Design and implement 3-bit Full adder with NOR gates using logisim simulator

****

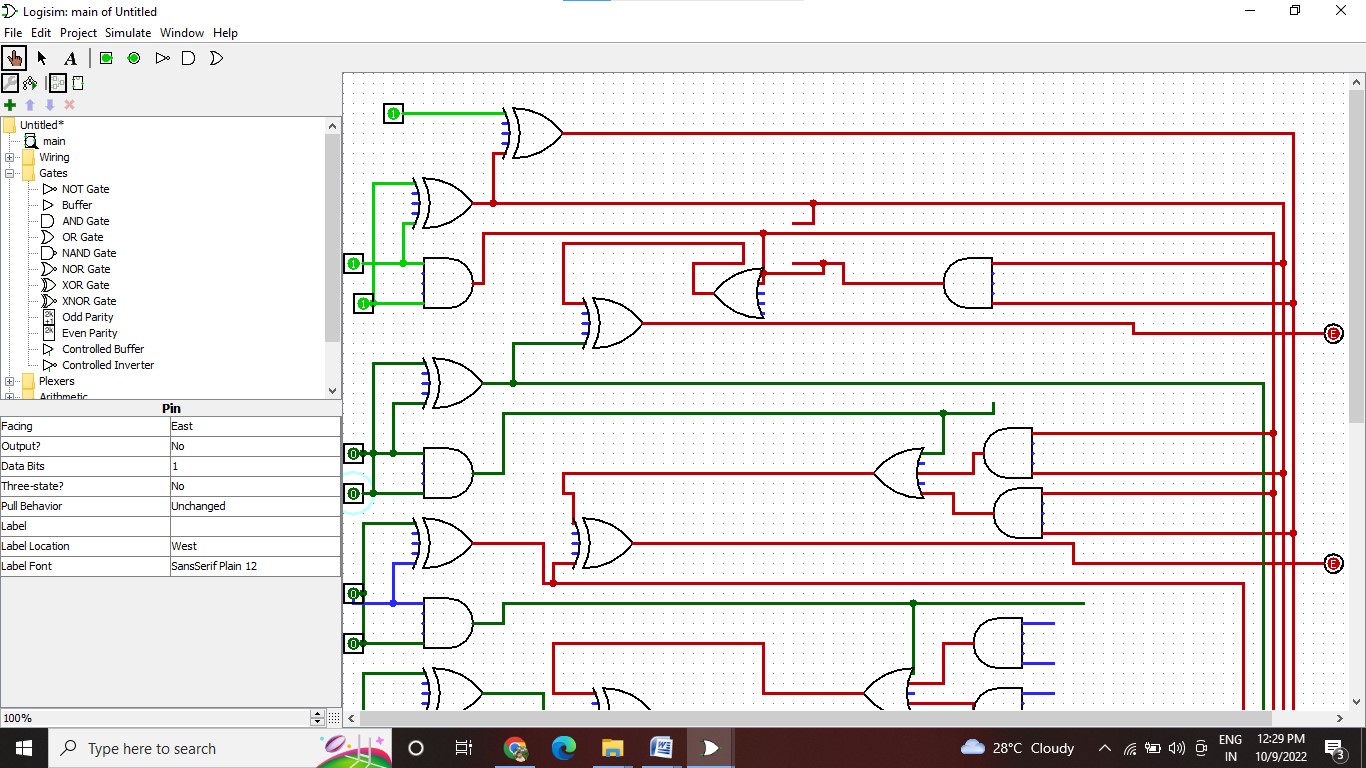
EXP : 45

Design and implement 4-bit ripple carry adder circuitusing logisim simulator

RIPPLE FACTOR



EXP 46:



EXP :47

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("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:

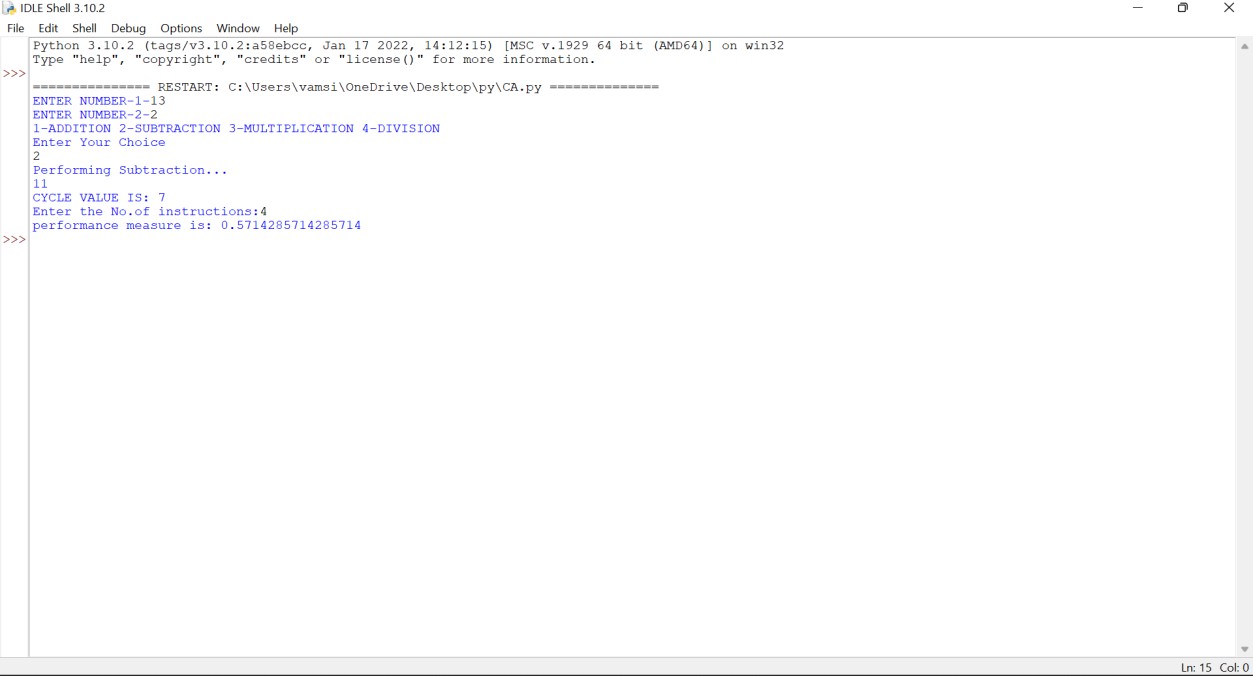
print("Enter Correct Input")

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:



EXP :49

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

. 