NAME:P.KALYAN

REG. NO:192110675

Course: CSA1288(computer architecture)

1.

8 BIT ADDITION

LDA 2050

MOV B,A

LDA 2052

ADD B

STA 2054

HLT

EX. 2

8 BIT SUBTRACTION

LDA 2050

MOV B,A

LDA 2052

SUB B

STA 2054

HLT

3.

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

4.

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

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

6.

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

7.

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

8.

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

**EXP 09**

LDA 2050

MOV B, A

LDA 2051

CMP B

JNC 2053

MOV A, B

STA 2060

HLT

10.

LHLD 2050

XCHG

LHLD 2052

MVI C,00

MOV A,E

SUB L

STA 2054

MOV A,D

SBB H

STA 2055

HLT

14.

def hex:

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

print("of the number is")

print(hex(n))

15.Binary to octal

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):

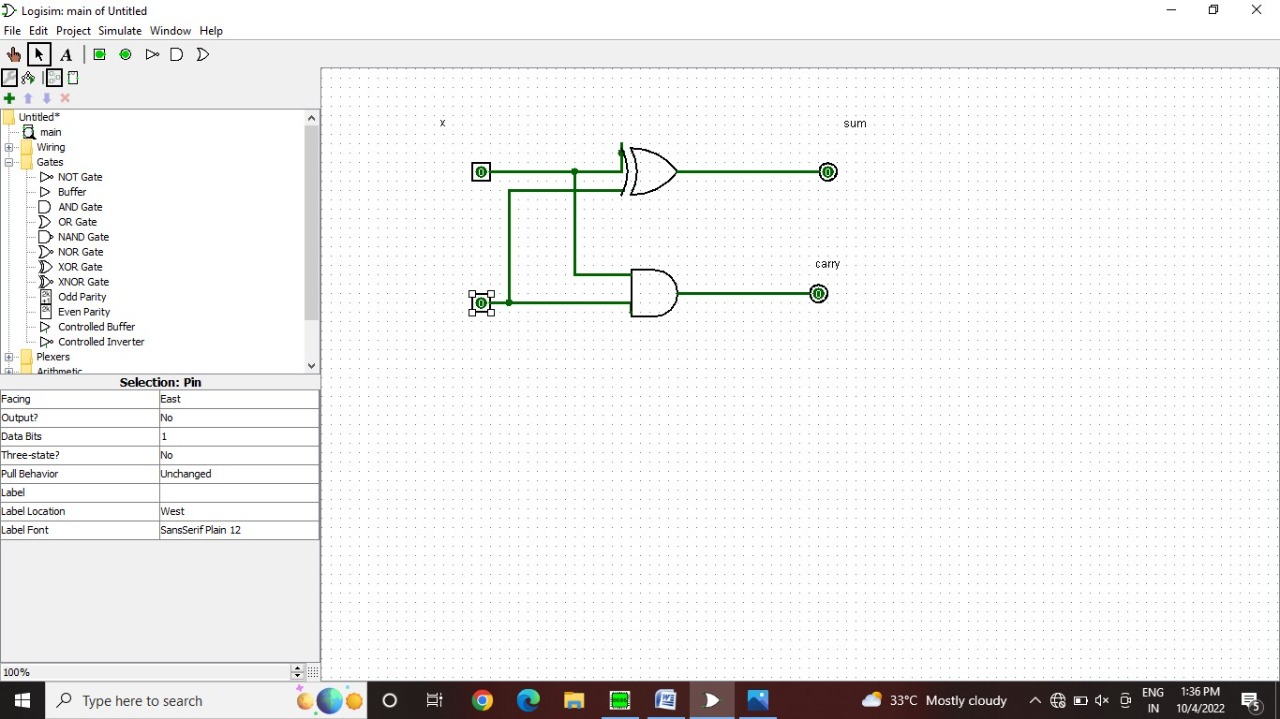
print(octalArray[j], end='')

binary = 1010

convert(binary)

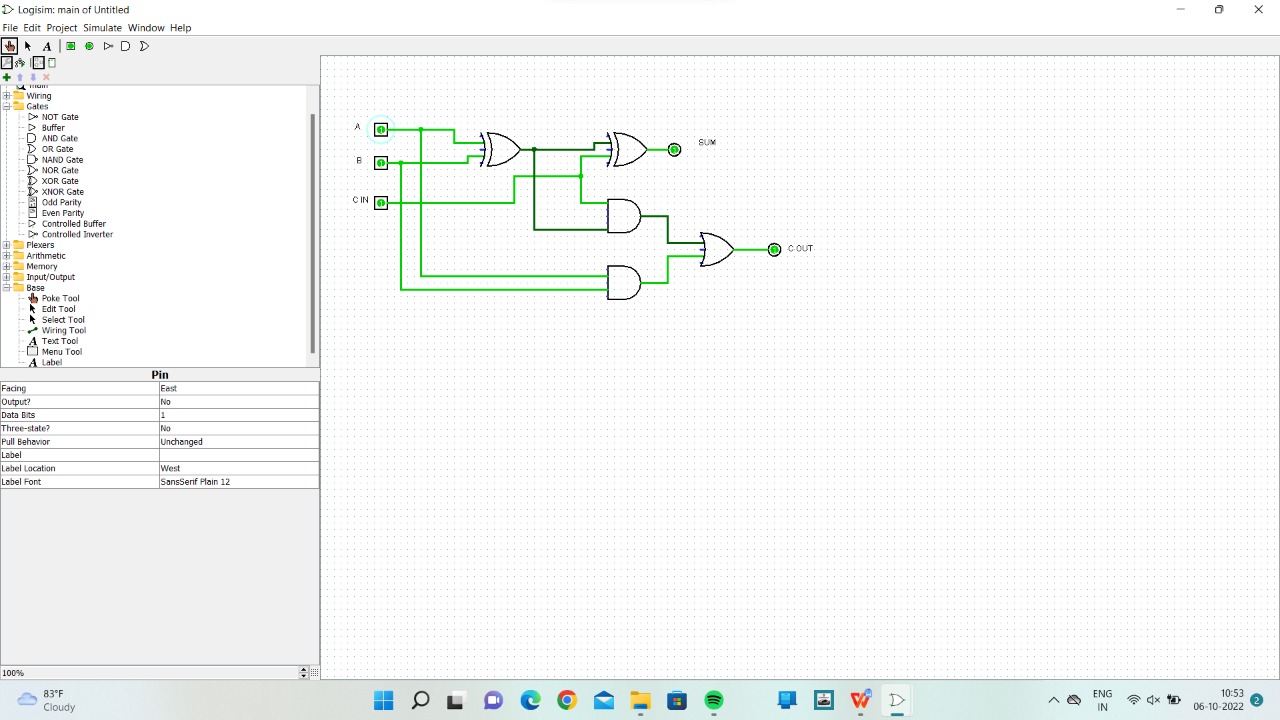
16.

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

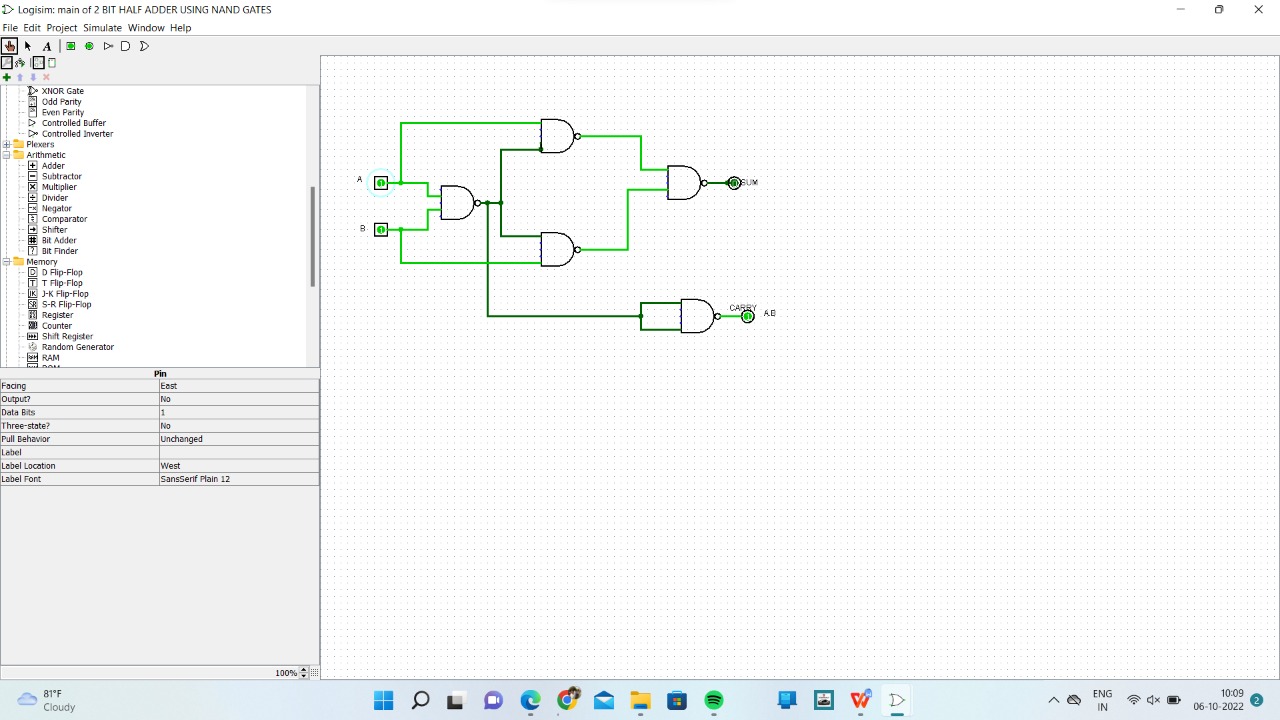


17.

THREE BIT FULL ADDER



18. 2 BIT HALF HADDER USING NAND GATES



21. BINARY TO HEXADECIMAL

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

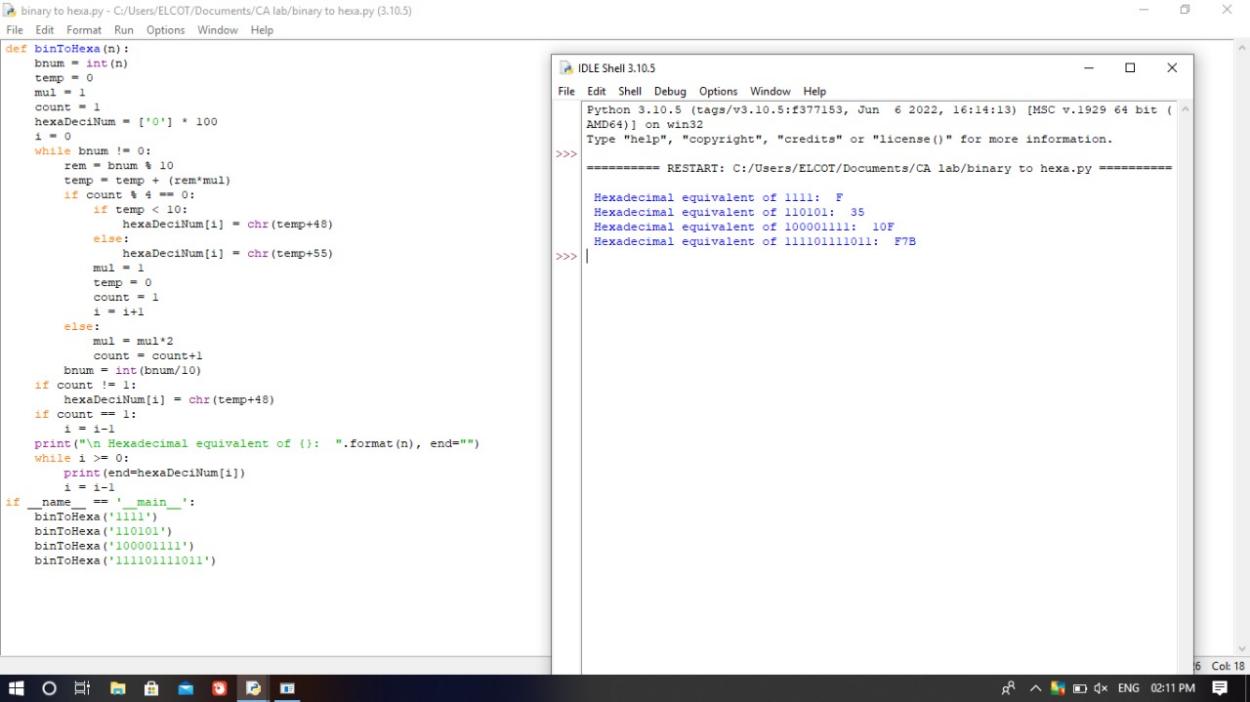
else:

mul = mul\*2

count = count+1

bnum = int(bnum/10)

output:



Exp :22

**TWO stage pipe line ADDITION**

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:

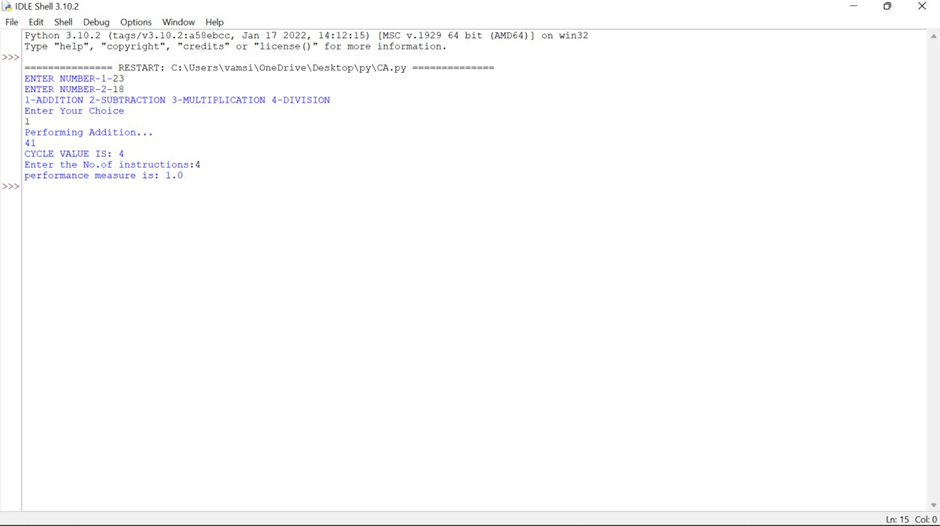
print("Enter Correct Input")

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 :23

2 STAGE pipe line Subtraction

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:

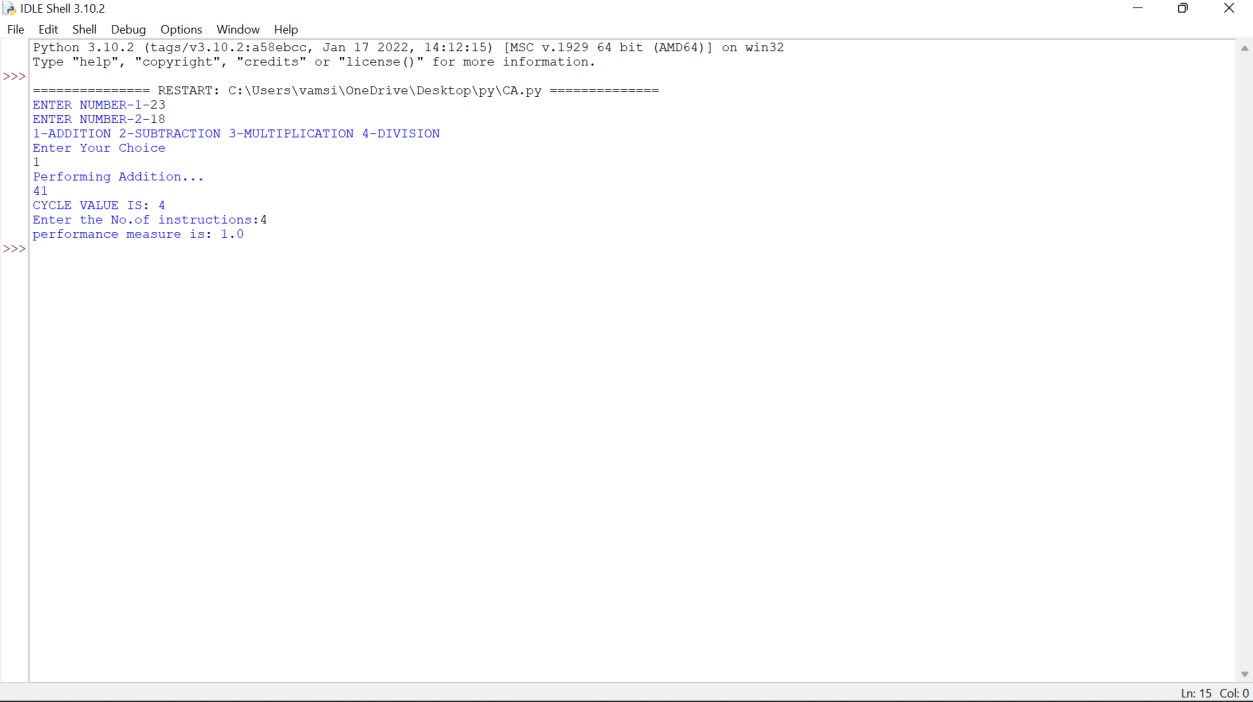
print("Enter Correct Input")

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

3 STAGE pipe line (AND)

counter=1

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

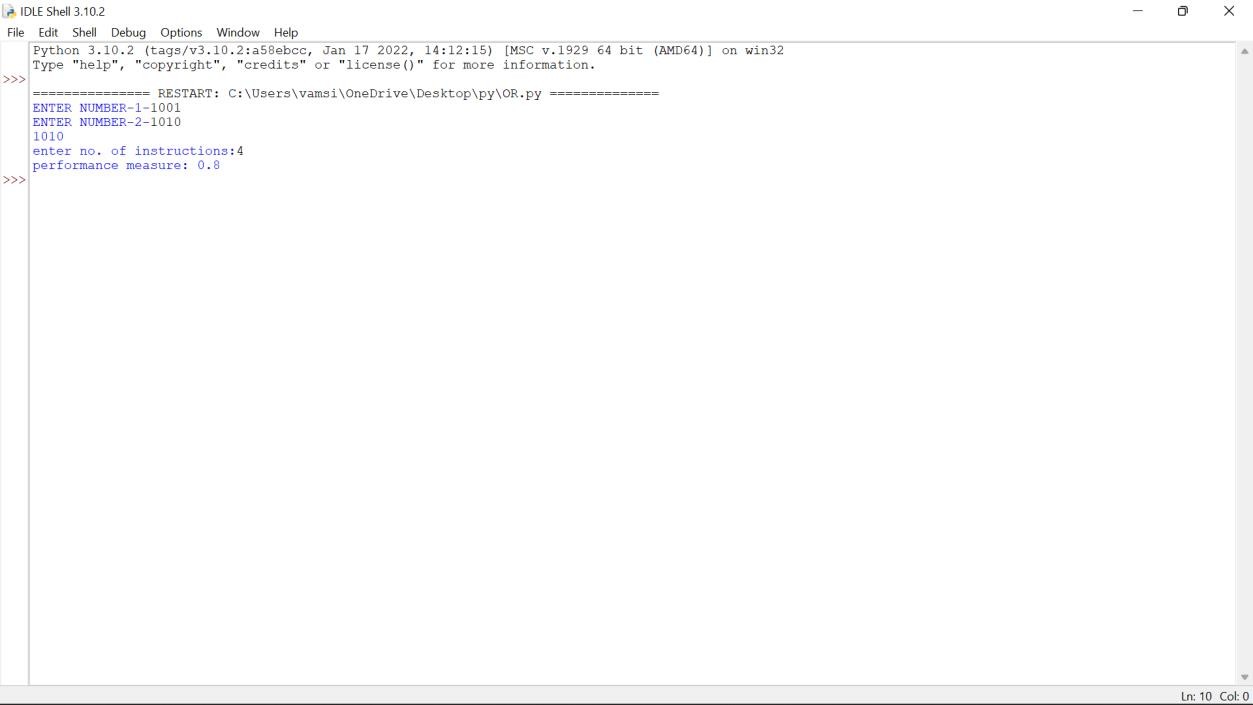
res= a and b counter=counter+1 print(res) counter=counter+2

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

performance\_measure=INS/counter

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

OUTPUT :



EXP : 25

3 STAGE pipe line (OR )

counter=1

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

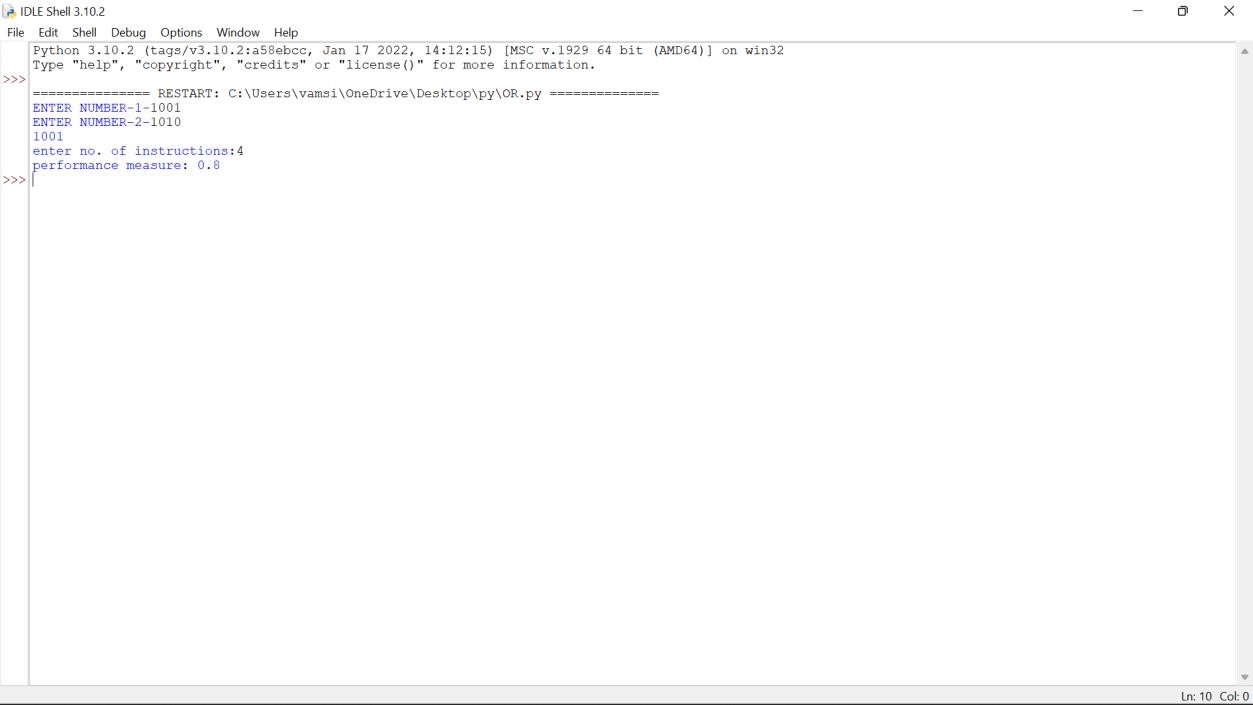
res= a or b counter=counter+1 print(res) counter=counter+2

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

performance\_measure=INS/counter

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

OUTPUT :



EXP :28

4 STAGE pipe line

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:

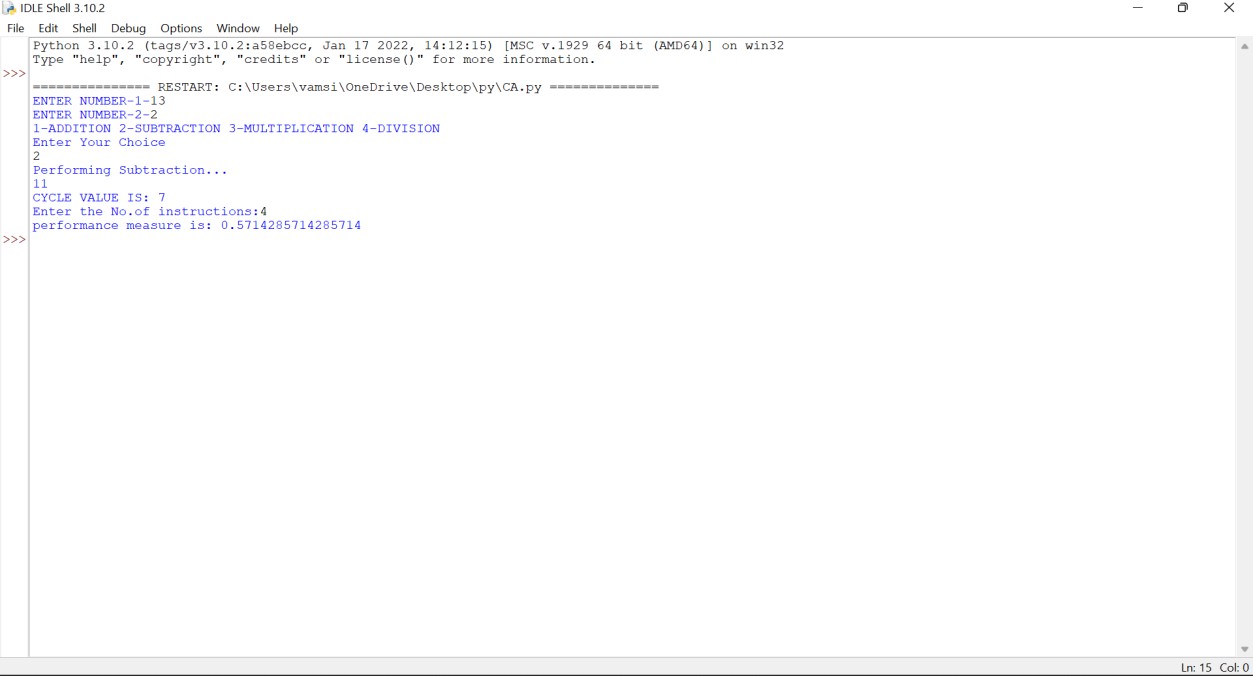
print("Enter Correct Input")

print(res) counter=counter+3

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

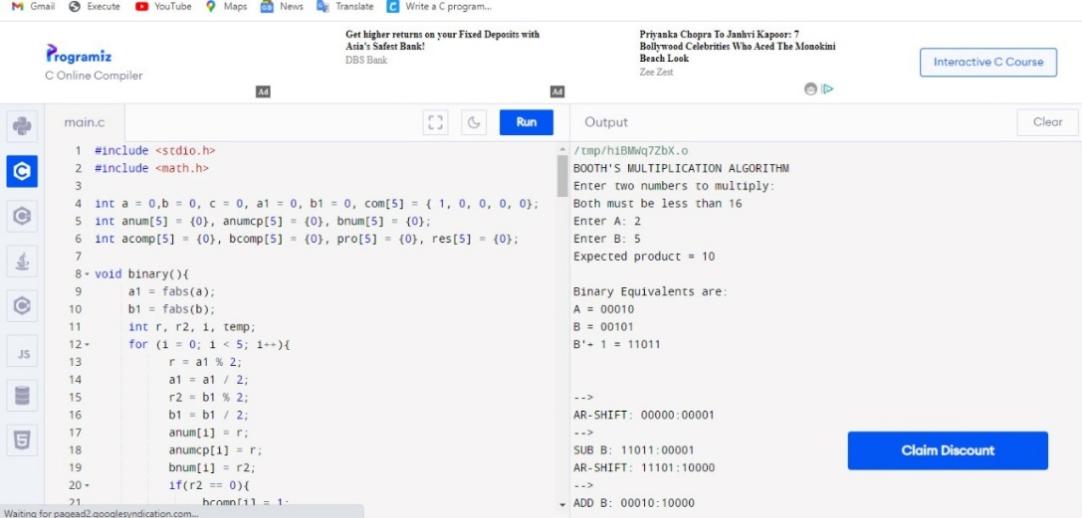
print("performance measure is:",performance\_measure)

OUTPUT :



EXP :29

OUTPUT



EXP :30

RESTORING DIVISION

#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

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

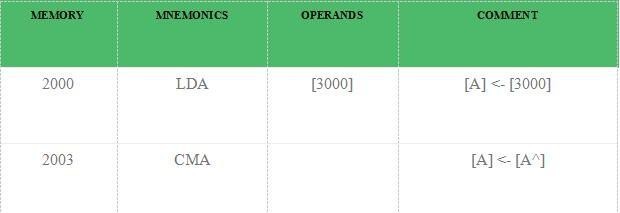
enter the number of cache misses:11

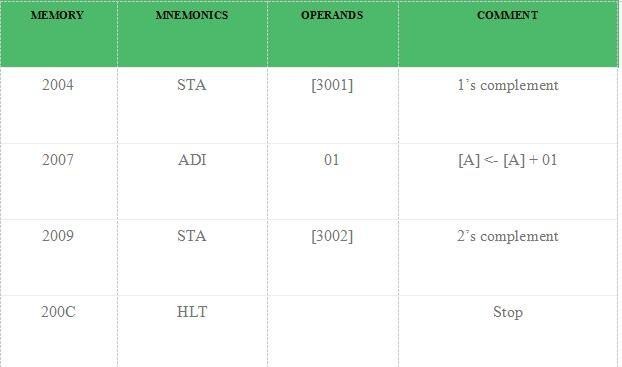
Cache Hit Ratio: 0.796296

Cache Miss Ratio: 0.20370

**EXP 32**

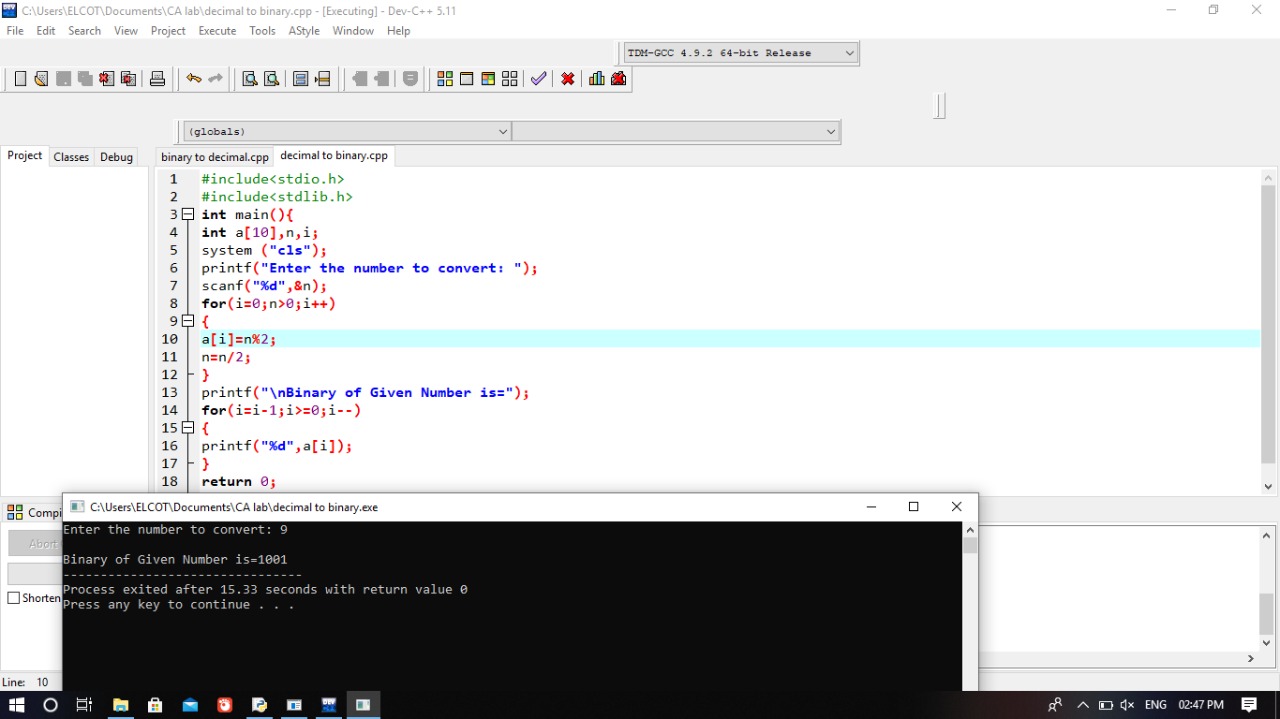
**EXP 33**





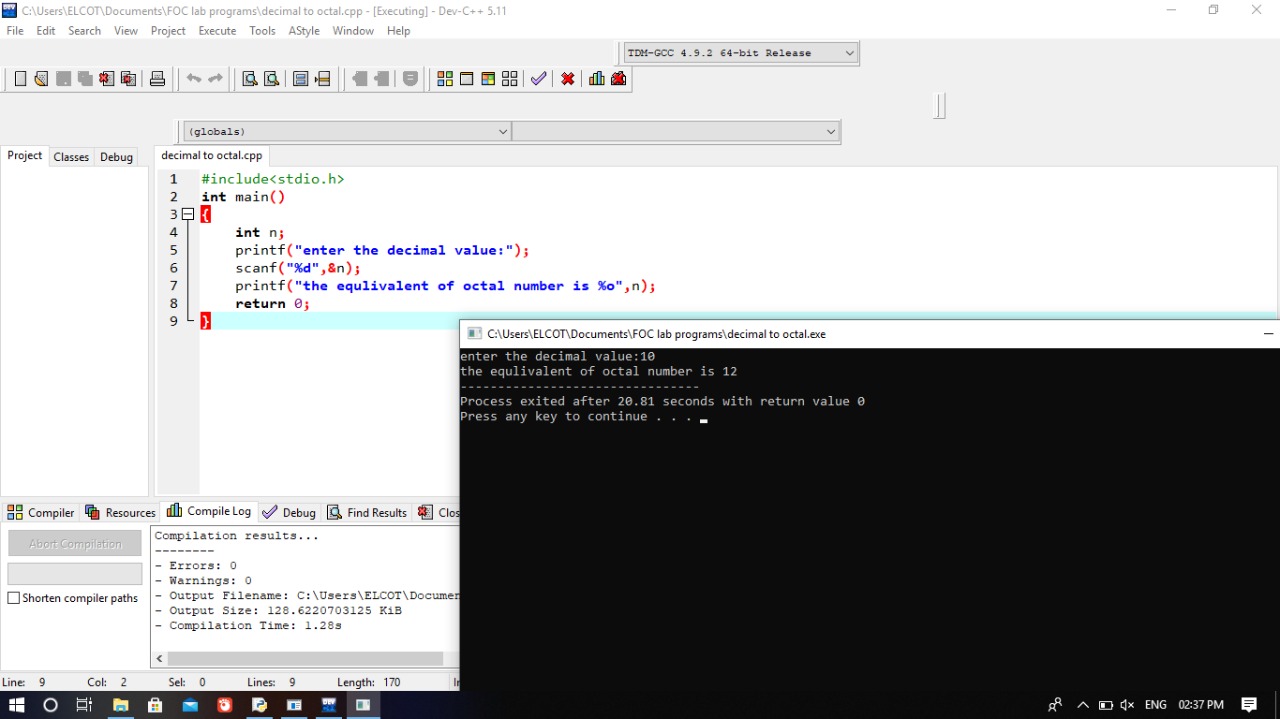
**EXP 36**

**DECIMAL TO BINARY**

****

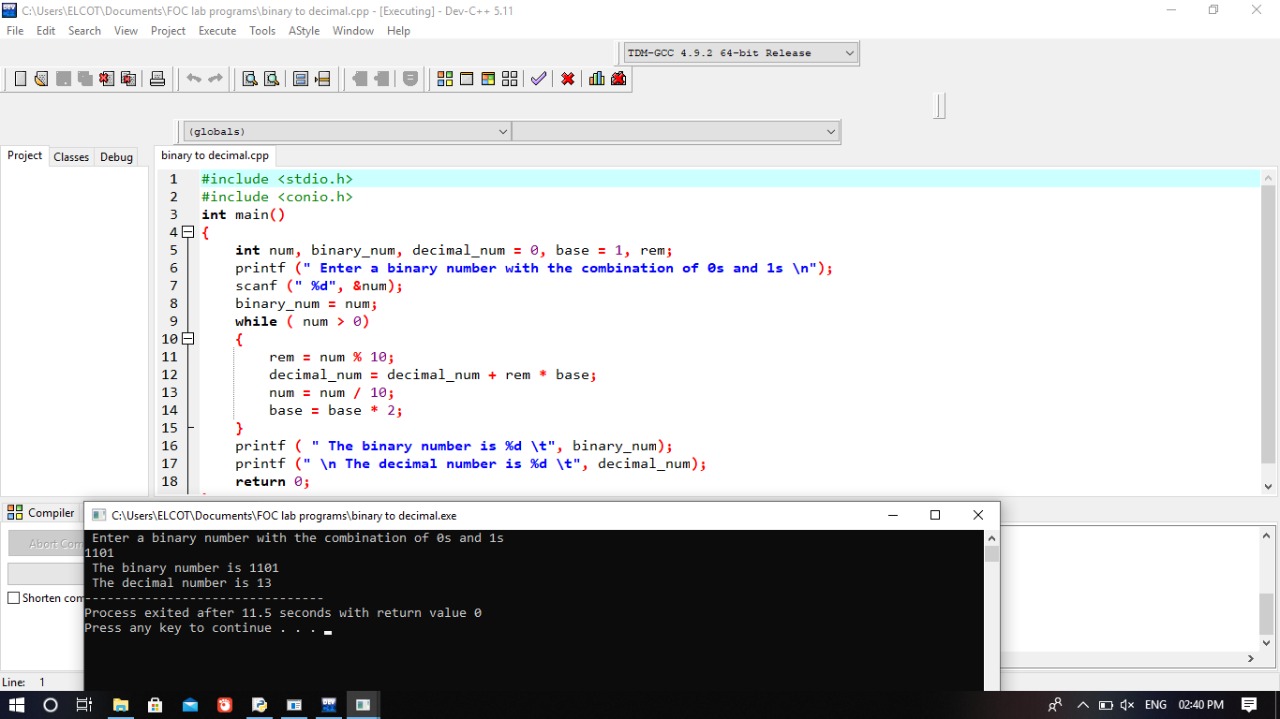
**EXP : 37**

**DECIMAL TO OCTAL**

****

**EXP 38**

**BINARY TO DECIMAL**

****

**EXP :39**

**CPU PERFORMANCE**

#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 :**

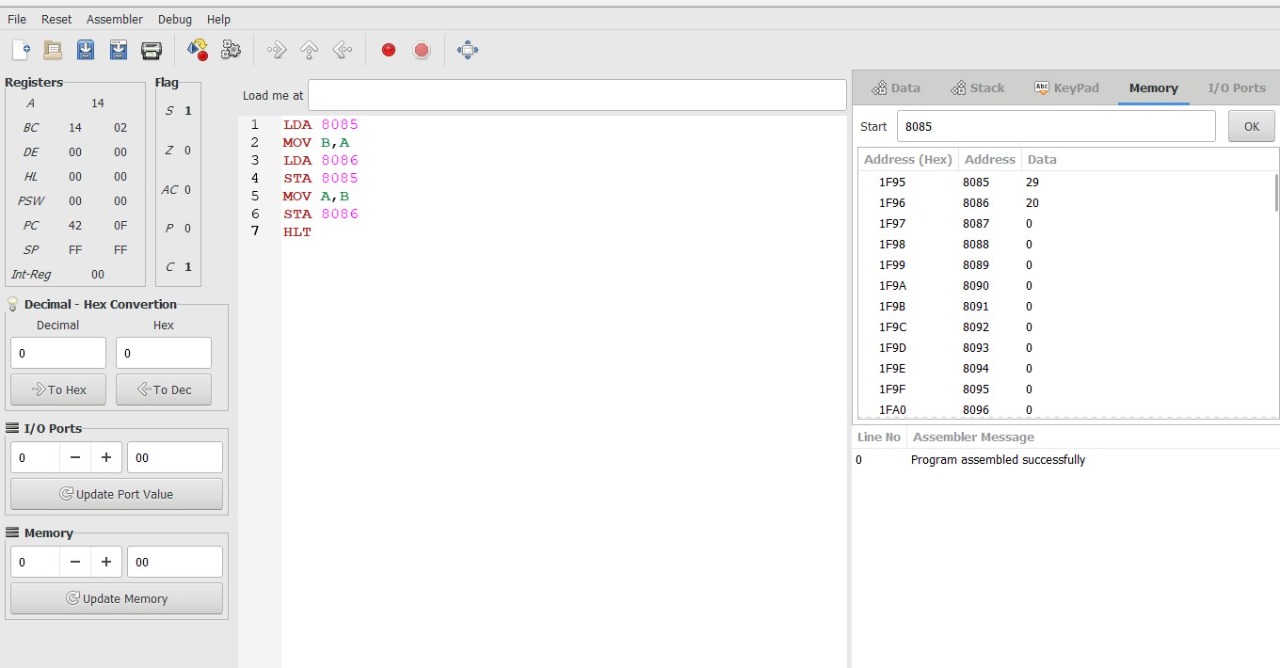
Enter the number of processors:3

Enter the Cycles per Instrcution of processor:1.5 Enter the clockrate in GHz:3

The CPU time is: 500.000000

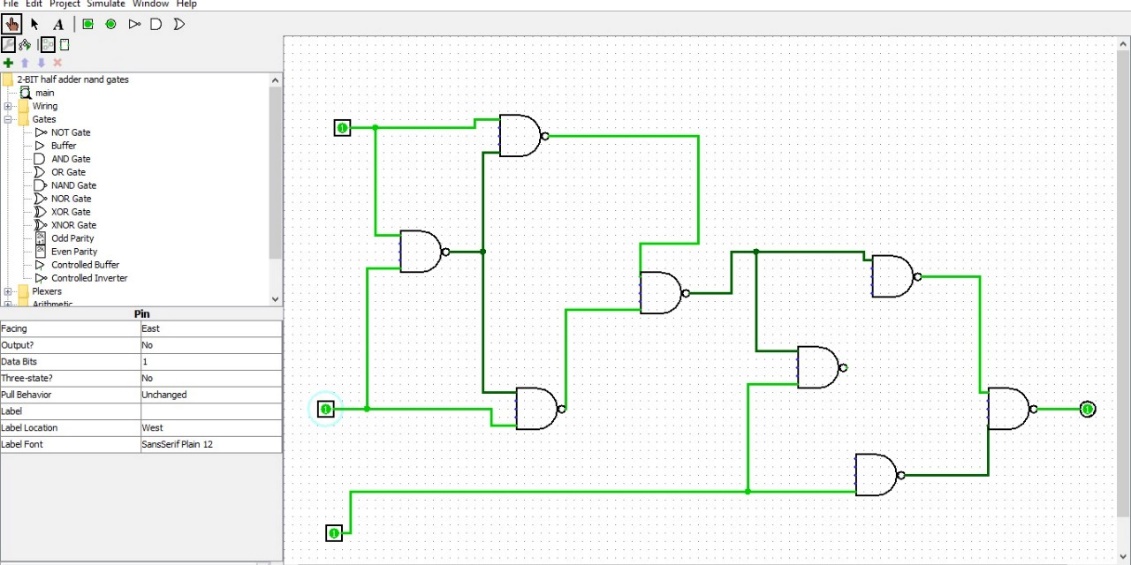
**EXP :40**

**SWAPPING**

****

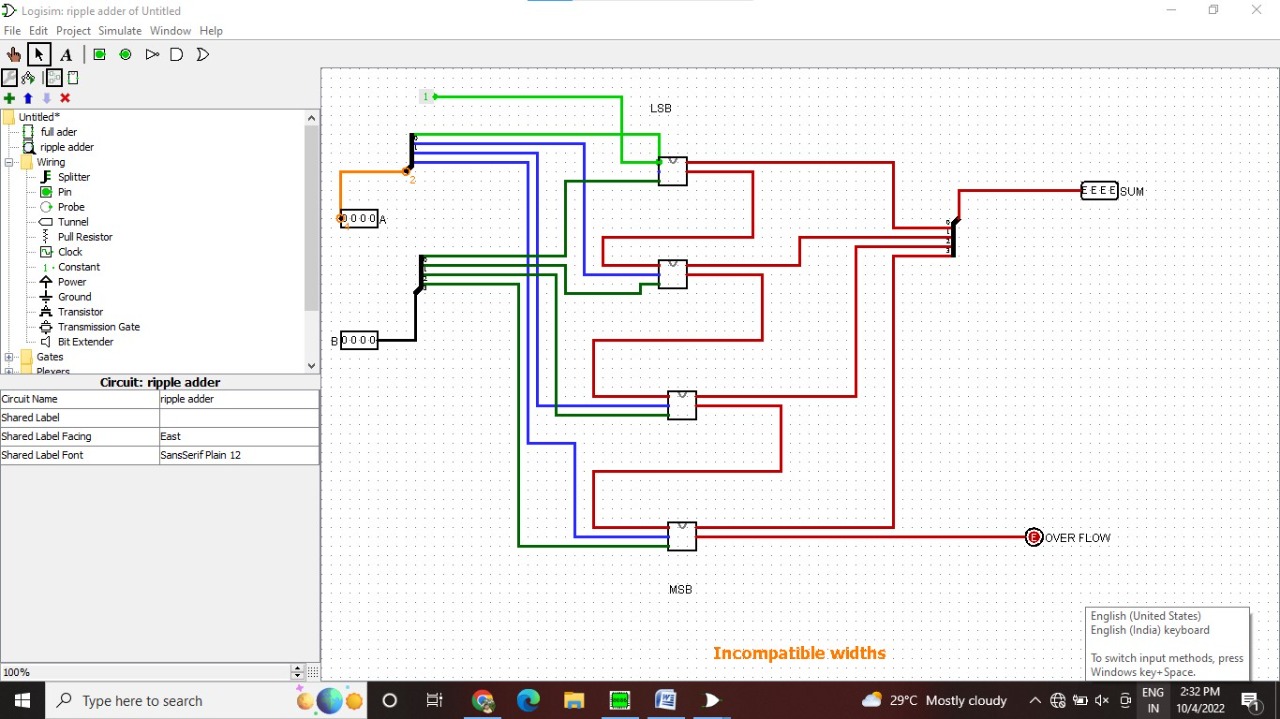
**EXP : 43**

**3 BIT FUUL ADDER WITH NAND GATE**

****

EXP : 45

RIPPLE FACTOR



EXP :47

4 STAGE PIPE LINE

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:

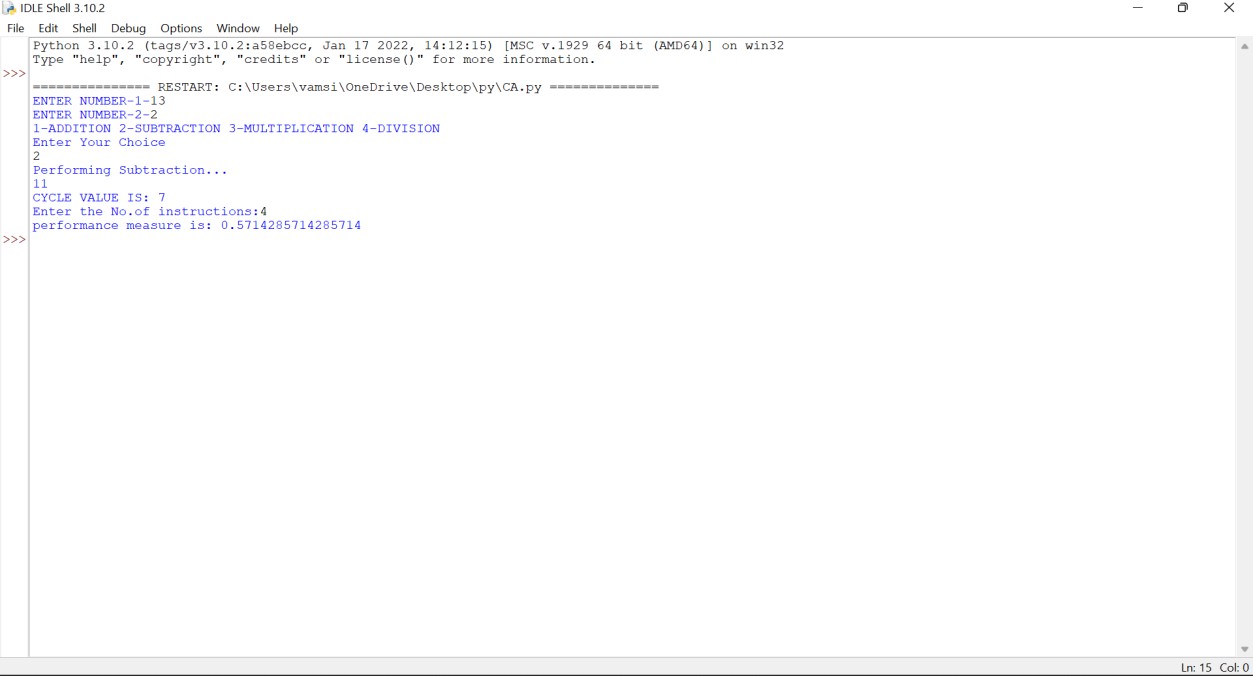
print("Enter Correct Input")

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

FIND ODD OR EVEN

