**GOPI KRISHNA. D**

**(192011065)**

**8-BIT ADDITION PROGRAM:**

LDA 8500

MOV B,A

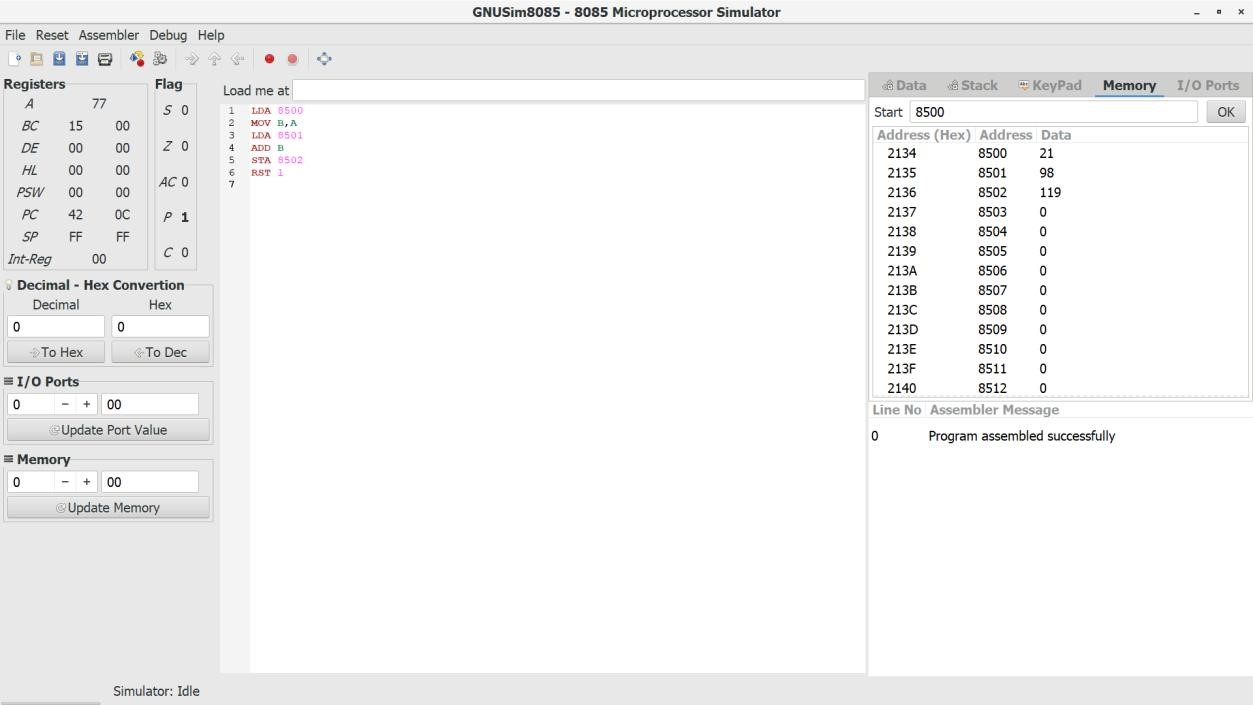
LDA 8501

ADD B

STA 8502

RST 1

**OUTPUT:**



**8-BIT SUBTRACTION PROGRAM:**

LDA 8500

MOV B,A

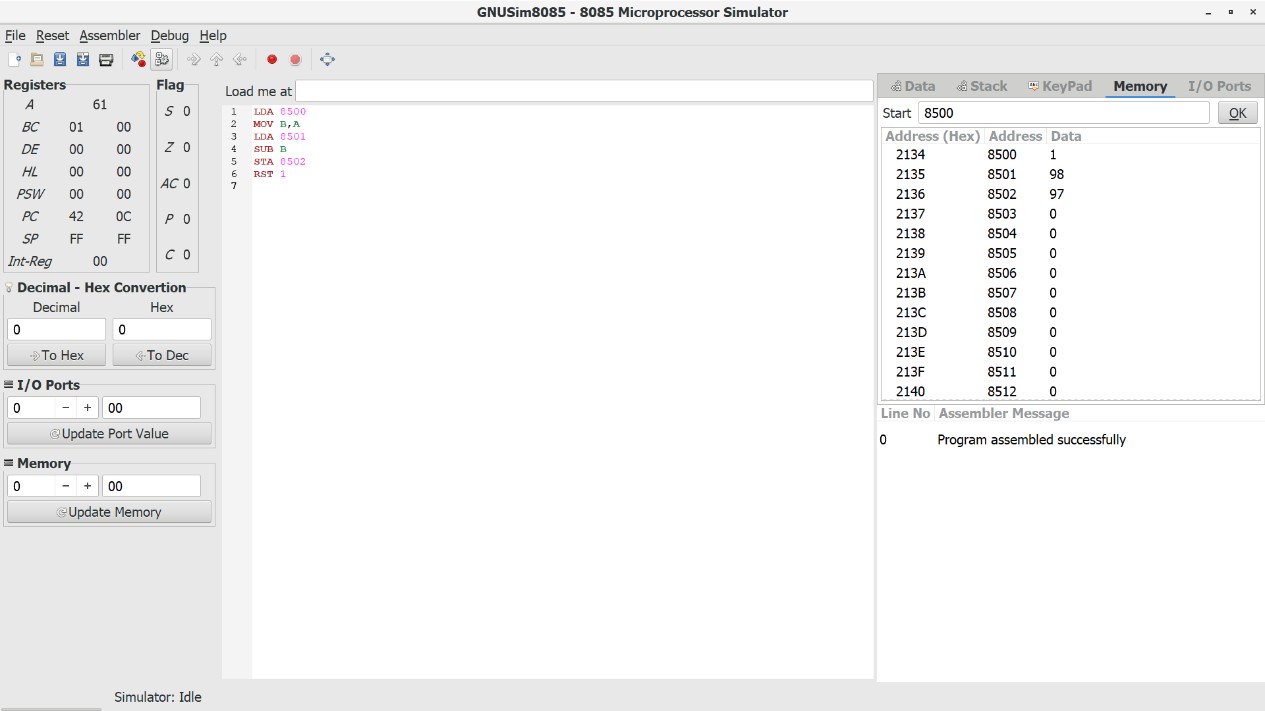
LDA 8501

SUB B

STA 8502

RST 1

**OUTPUT:**



# 8-BIT MULTIPLICATION

**PROGRAM:**

LDA 8500

MOV B,A

LDA 8501

MOV C,A

CPI 00

JZ LOOP

XRA A

LOOP1: ADD B

DCR C

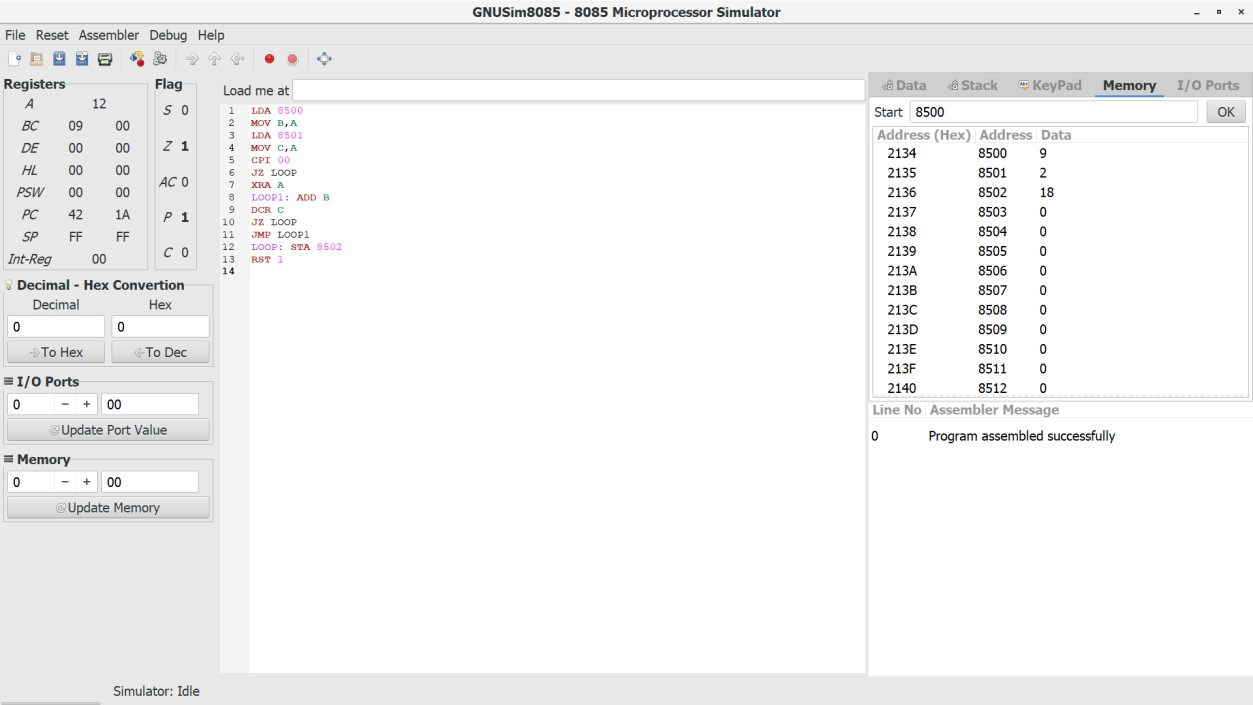
JZ LOOP

JMP LOOP1

LOOP: STA 8502

RST 1

**OUTPUT:**



# 8-BIT DIVISION

LDA 8501

MOV B,A

LDA 8500

MVI C,00

LOOP1: CMP B

JC LOOP

SUB B

INR C

JMP LOOP1

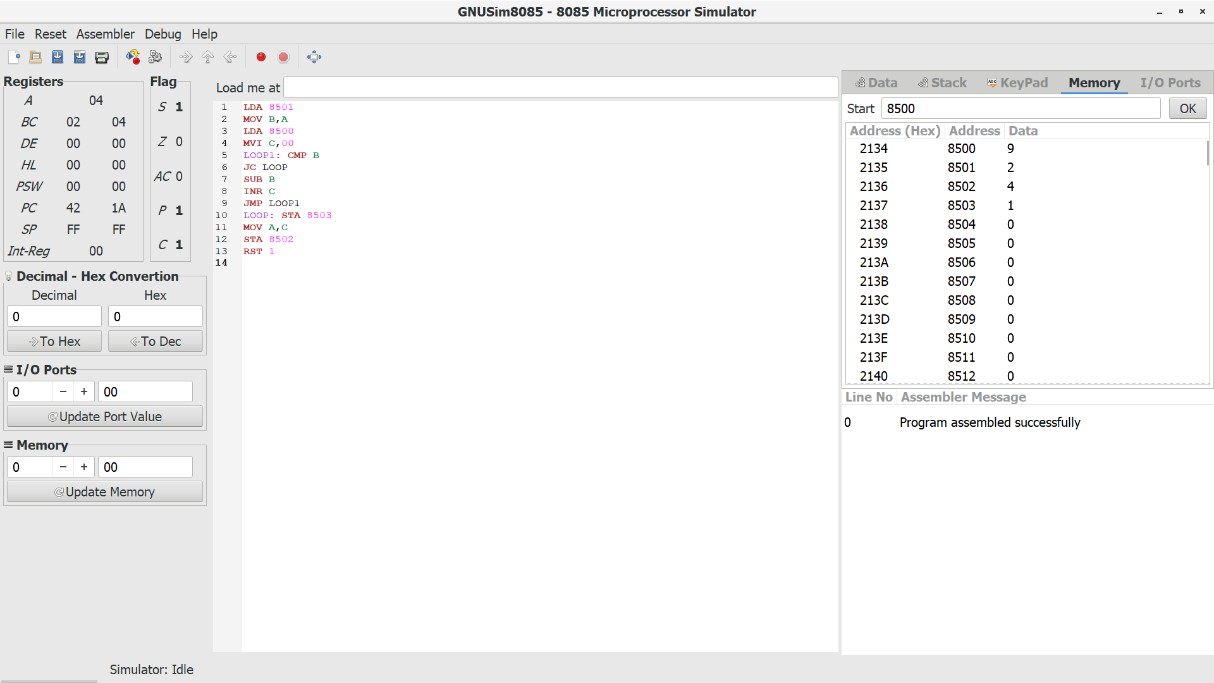
LOOP: STA 8503

MOV A,C

STA 8502

RST 1

**OUTPUT:**



**SWAPPING OF TWO NUMBERS PROGRAM:**

LDA 8085

MOV B,A

LDA 8086

STA 8085

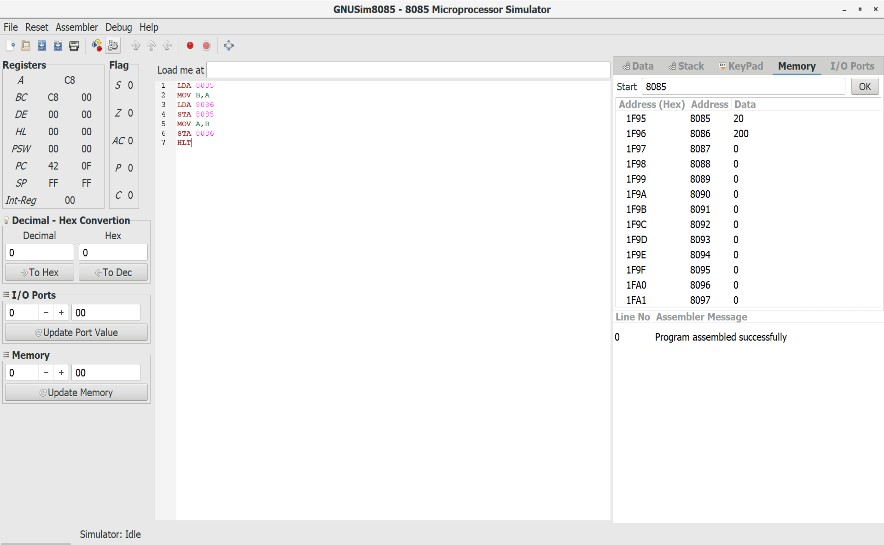
MOV A,B

STA 8086

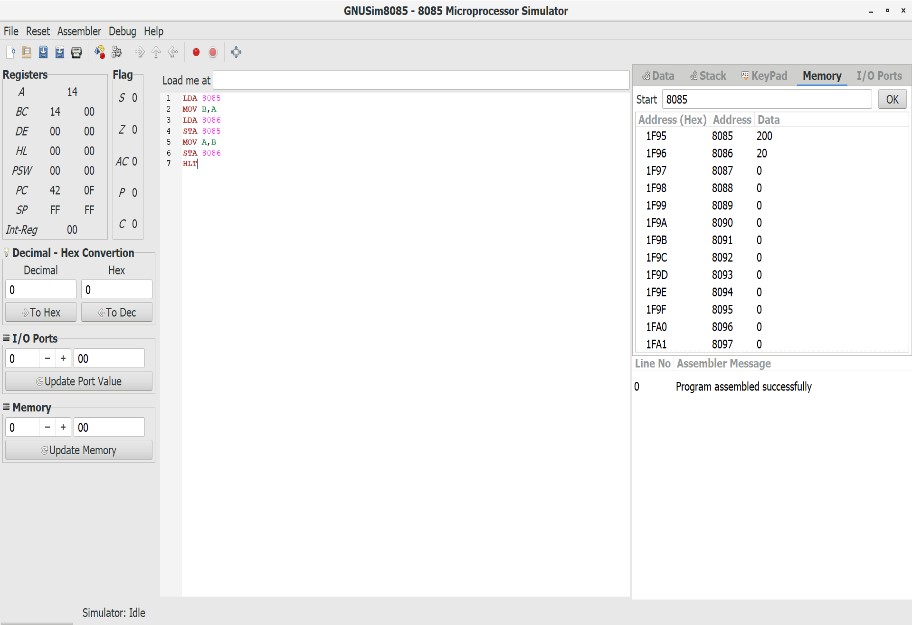
HLT

**OUTPUT:**

**(BEFORE SWAP)**



**(AFTER SWAP)**



## TWO STAGE PIPELINING

-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:

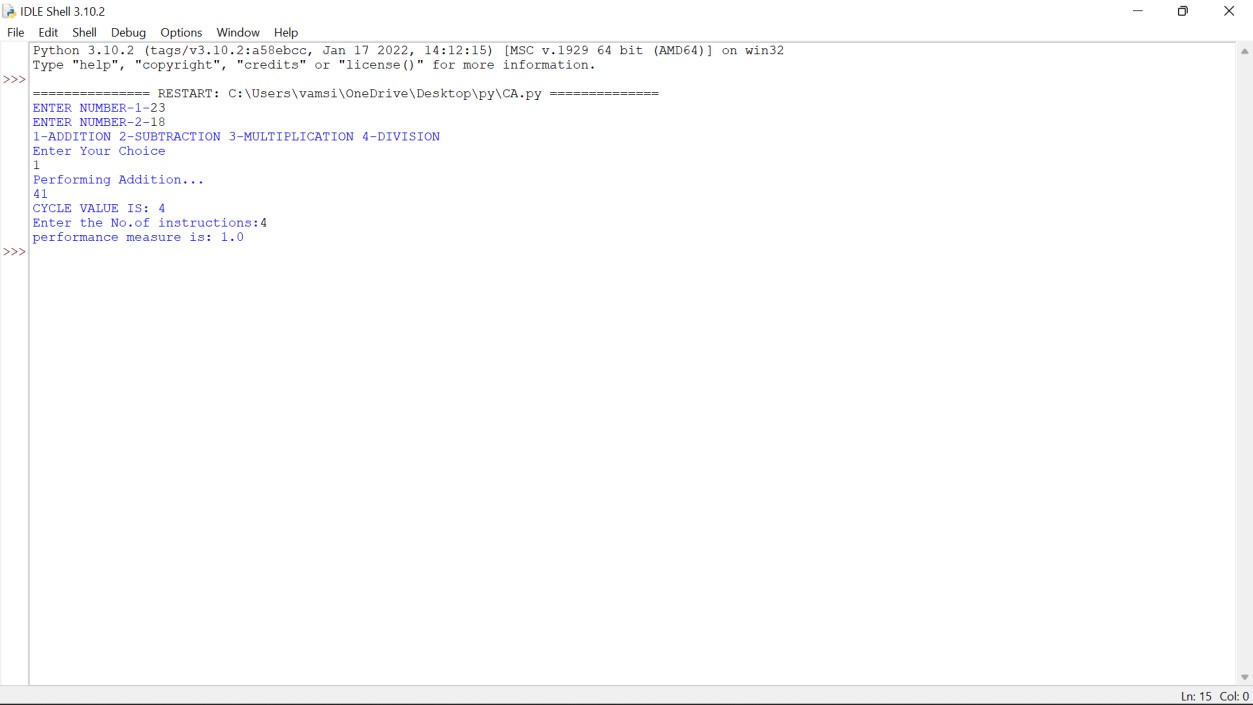
print("Denominator can't be Zero") print("Performing Division") res=a/b counter=counter+1 if choice>=5:

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:**



## THREE STAGE PIPELINE (AND)

-1-"))

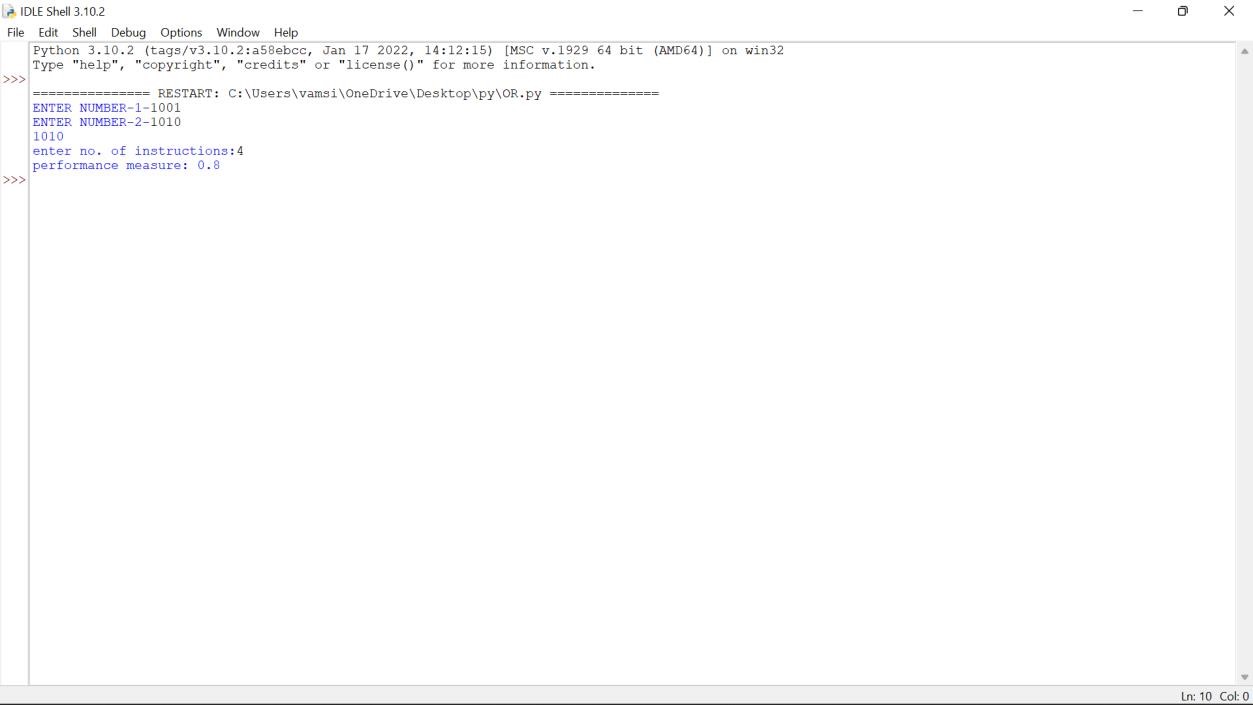
counter=counter+1

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:**



## THREE STAGE PIPELINE (OR)

-1-"))

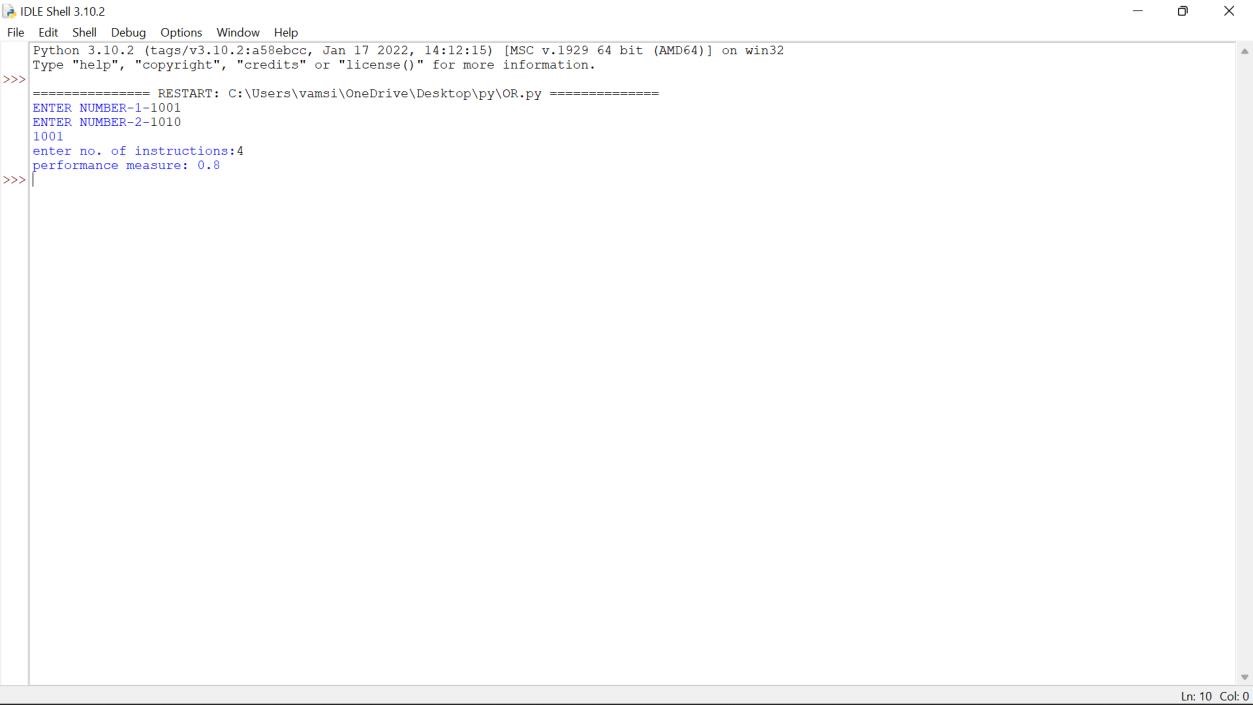
counter=counter+1

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:**



## FOUR STAGE PIPELINE

-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:

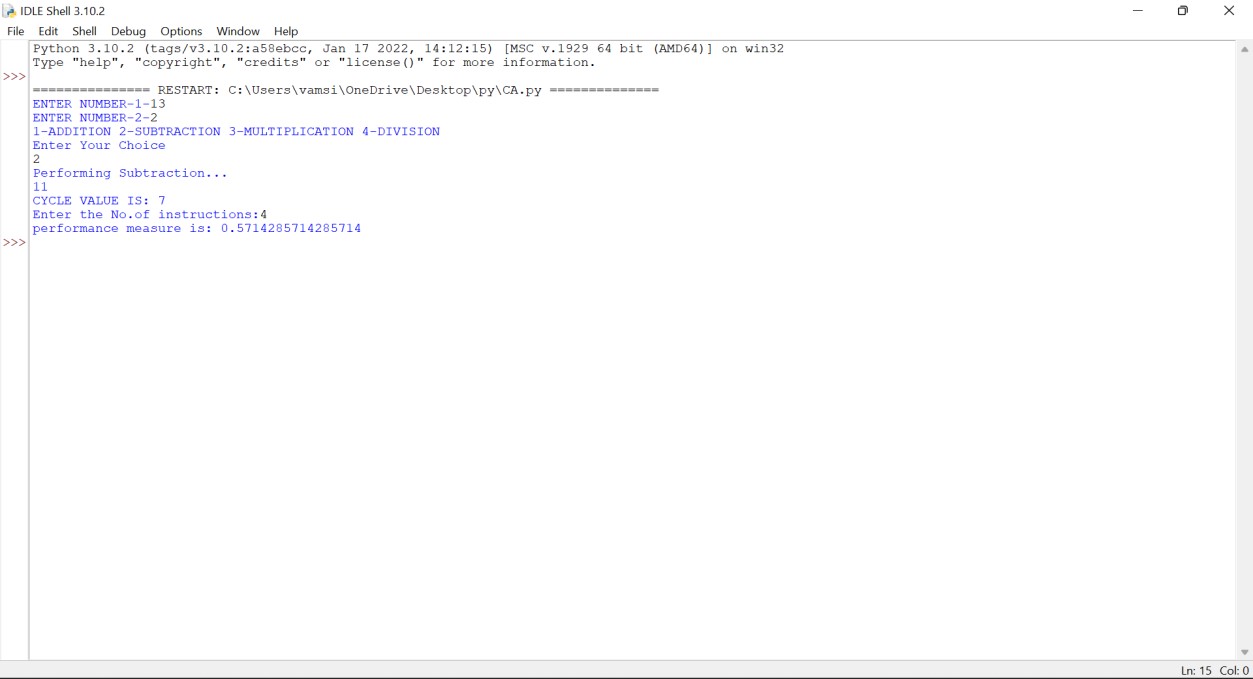
print("Enter Correct Input")

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:**



**16 BIT ADDITION(8086)**

MOV SI,1200H

LODSW

MOV BX,AX

LODSW

ADD BX,AX

MOV DI,1300H

MOV [DI],BX

HLT

**INPUT:**

1200 13H 1201 13H

1202 14H 1203 14H

**OUTPUT:**

1. 27H
2. 27H

**16 BIT SUBTRACTION PROGRAM:**

MOV SI,1200H

LODSW

MOV BX,AX

LODSW

SUB BX,AX

MOV DI,1300H

MOV [DI],BX

HLT

**INPUT:**

1200 08H 1201 08H

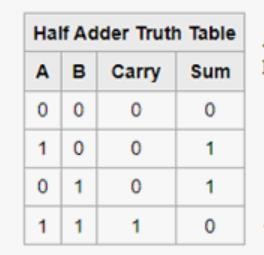
1202 04H 1203 04H

**OUTPUT:**

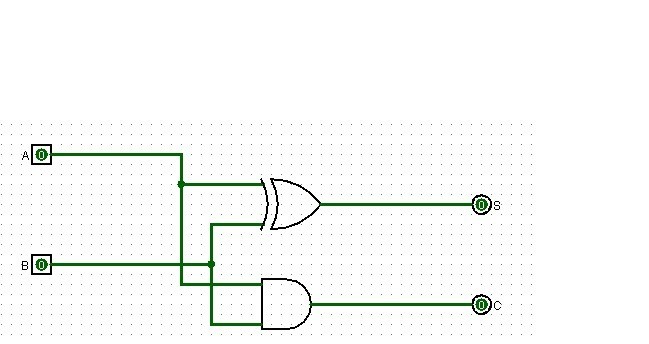
1. 04H
2. 04H

## HALF ADDER

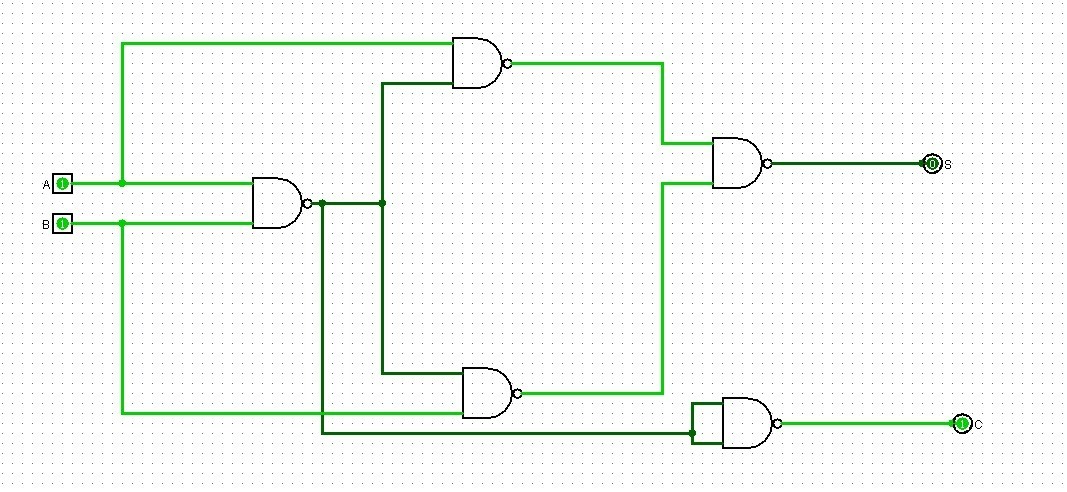
**TRUTH TABLE:**



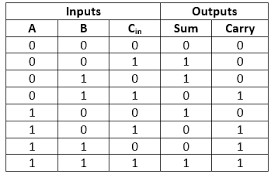
**LOGISM CIRCUIT DIAGRAM:**



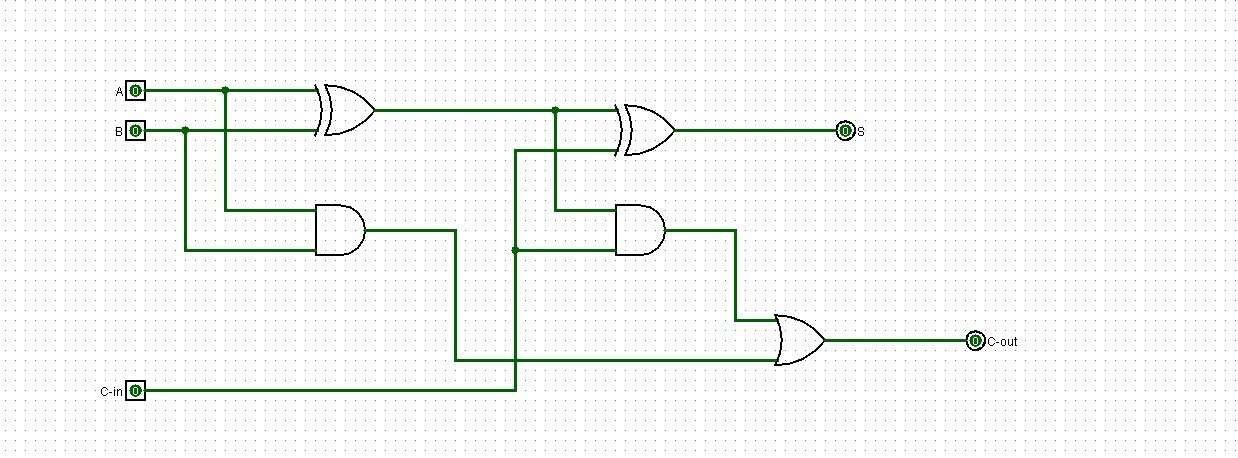
**LOGISM CIRCUIT DIAGRAM (USING NAND GATES)**



**FULL ADDER TRUTH TABLE:**



**LOGISM CIRCUIT DIAGRAM:**



**LOGISM CIRCUIT DIAGRAM (USING NAND GATE):**

