***1.8 BIT ADDITION***

LDA 8500

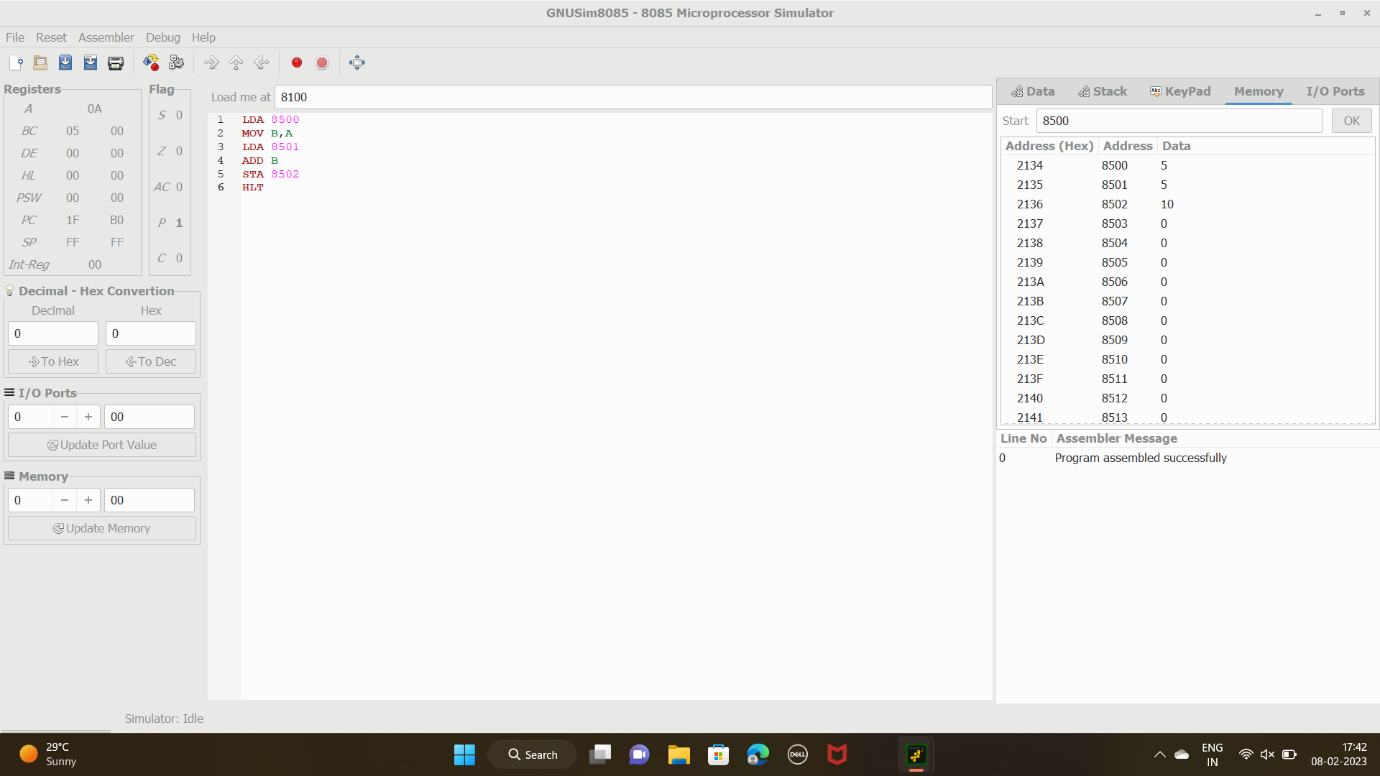
MOV B,A

LDA 8501

ADD B

STA 8502

HLT

******

***2.8 BIT SUBTRACTION***

LDA 9001

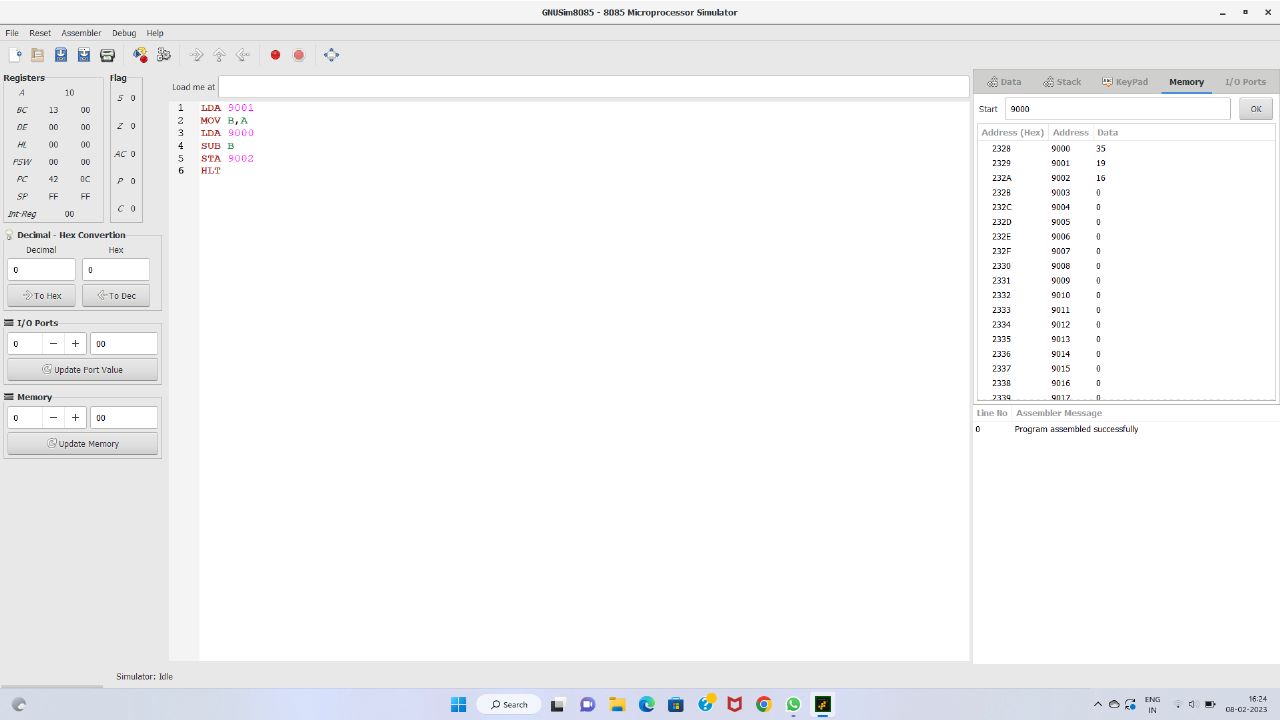
MOV B,A

LDA 9000

SUB B

STA 9002

HLT



***3. 16 BIT ADDITION***

LHLD 9000

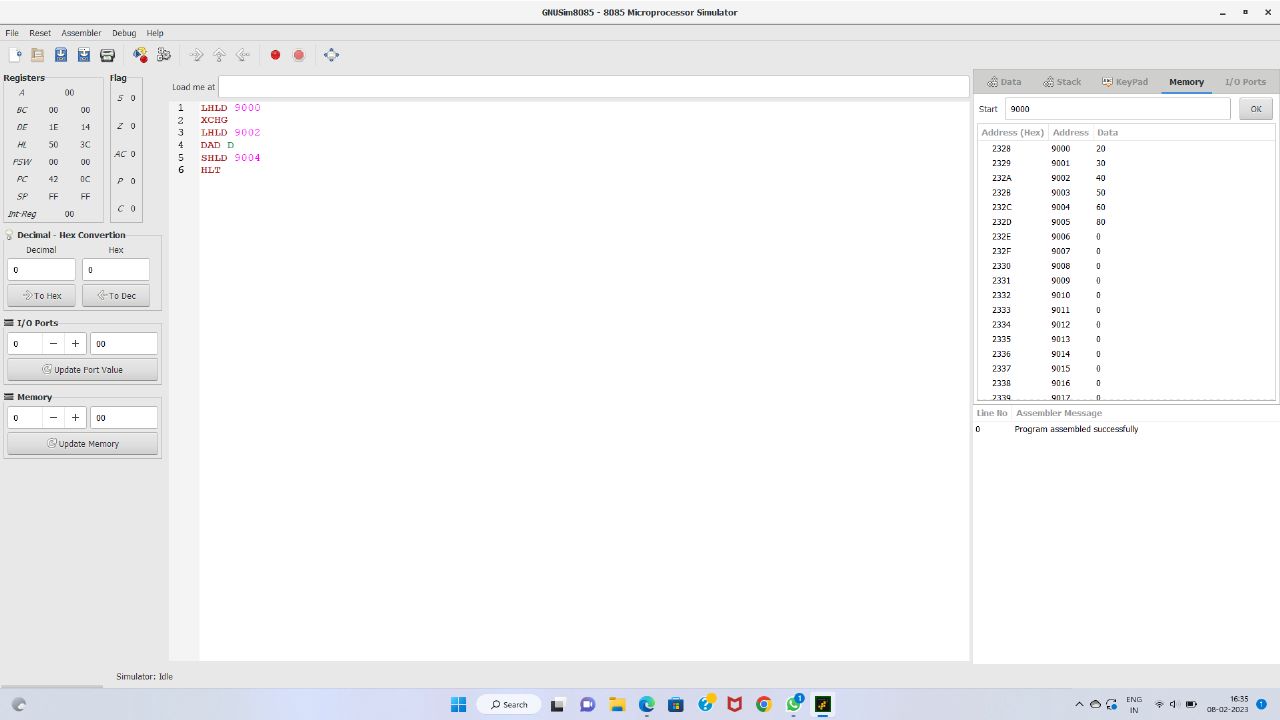
XCHG

LHLD 9002

DAD D

SHLD 9004

HLT



***4.16 BIT SUBTRACTION***

start: nop

LHLD 2501H

XCHG

LHLD 2503H

MOV A,L

SUB E

MOV L,A

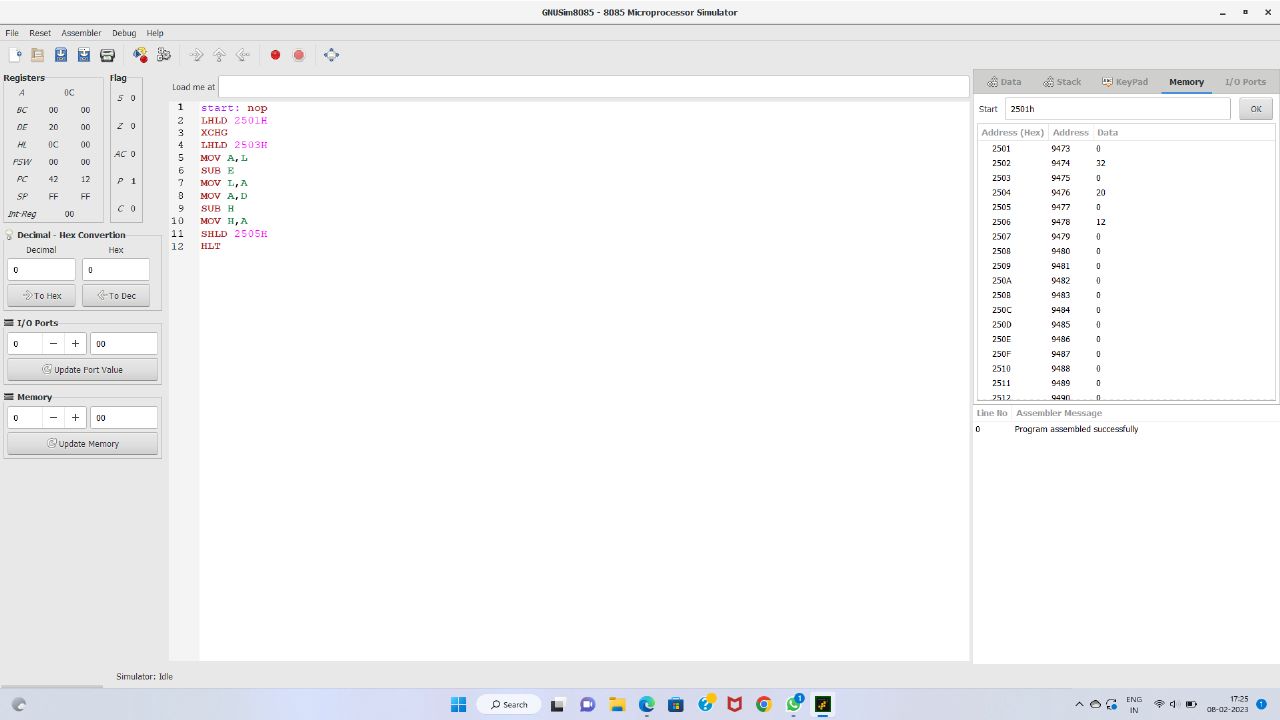
MOV A,D

SUB H

MOV H,A

SHLD 2505H

HLT



***5.8 BIT MULTIPLICATION***

LXI H,1100

MOV B,M

MVI A,00

MOV C,A

INX H

cont: ADD M

JNC skip

INR C

skip: DCR B

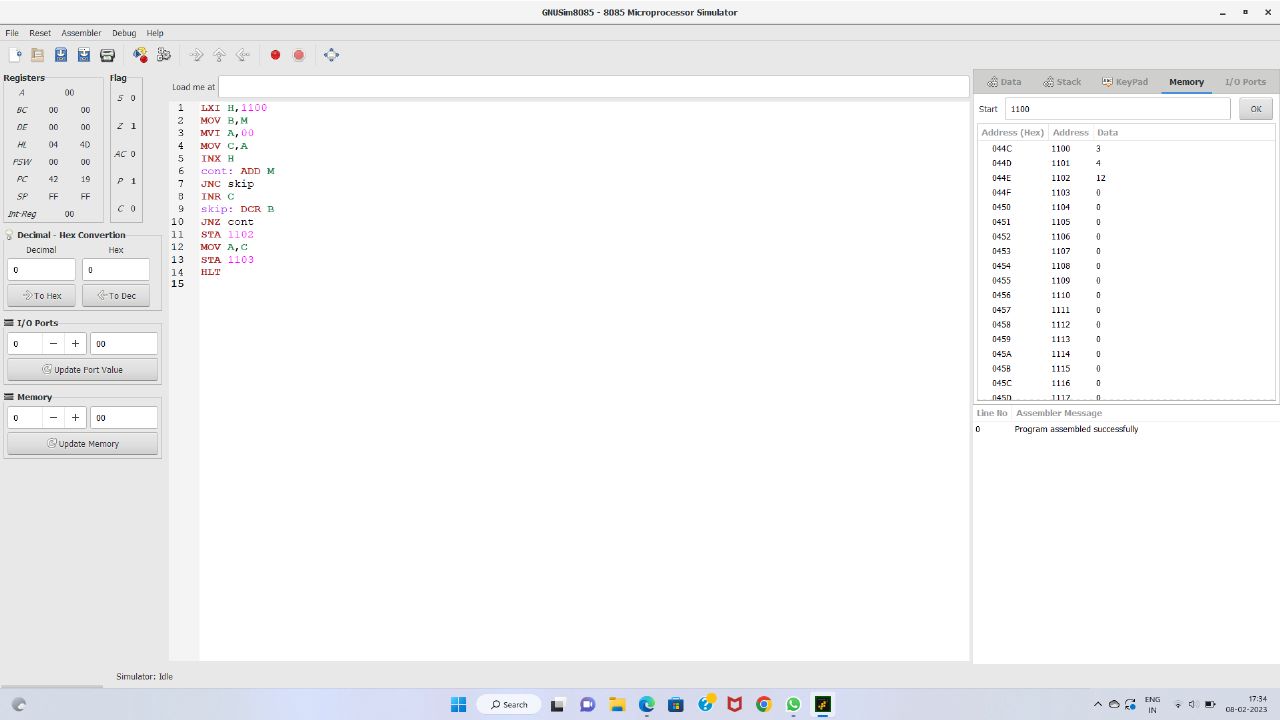
JNZ cont

STA 1102

MOV A,C

STA 1103

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

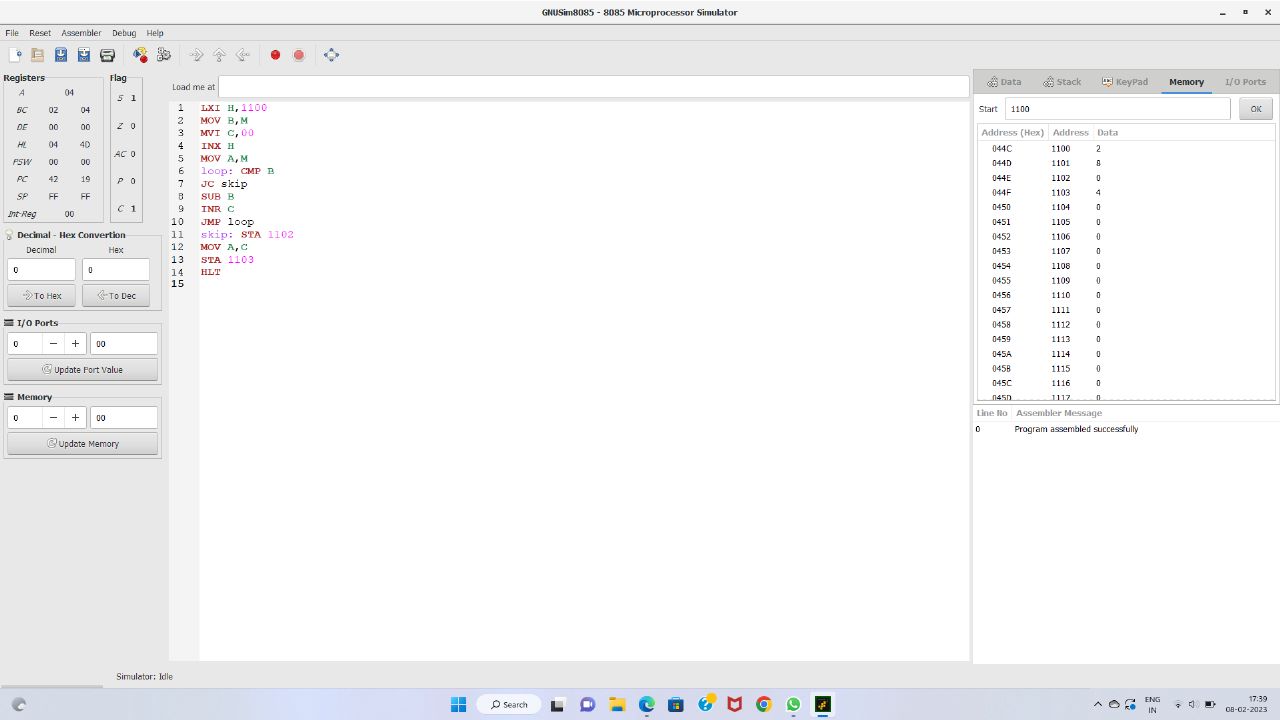
JMP loop

Skip: STA 1102

MOV A,C

STA 1103

HLT



***7.16 BIT MULTIPLICATION***

LHLD 2000

XCHG

LHLD 2020

MOV C,H

MVI A,00H

Loop: ADD D

DCR C

JNZ loop

MOV H,A

MOV B,L

MVI A,00H

lp: ADD E

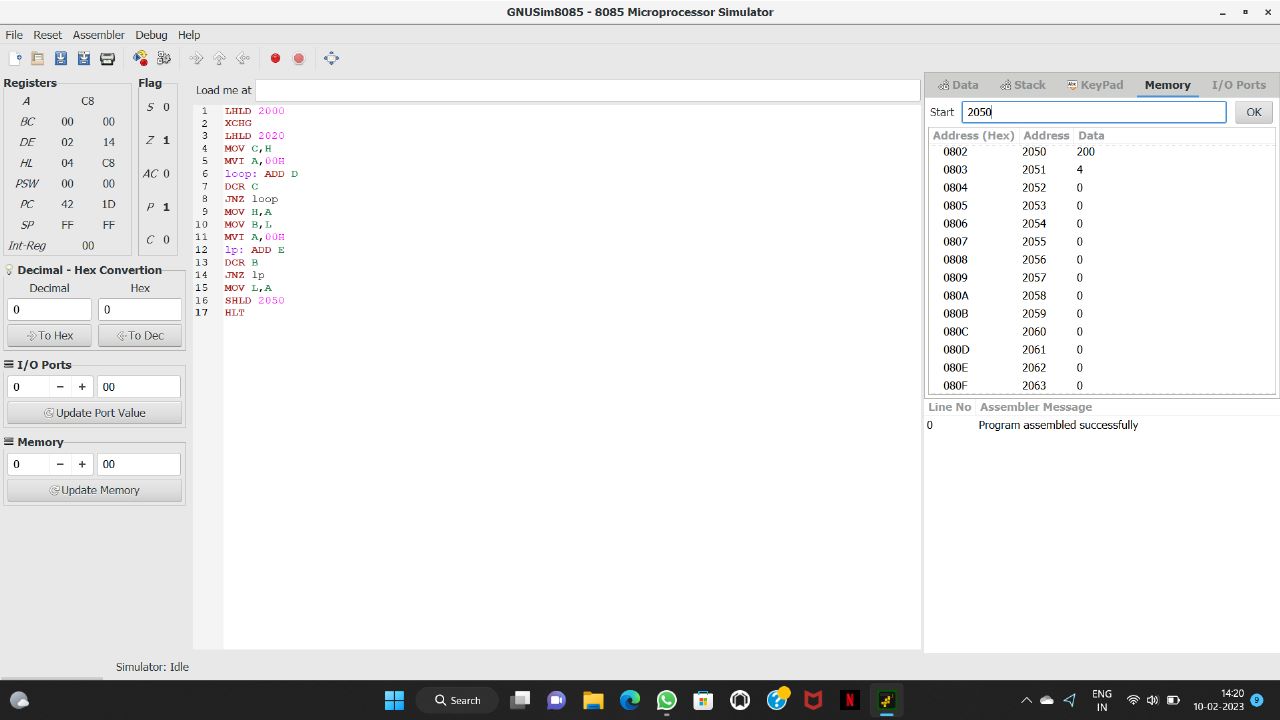
DCR B

JNZ lp

MOV L,A

SHLD 2050

HLT



***8.16 BIT DIVISION***

LHLD 2000

XCHG

LHLD 2020

MOV A,D

MOV B,H

MVI C,00H

loop: INR C

SUB B

JNZ loop

MOV H,C

MOV A,E

MOV B,L

MVI C,00H

lp: INR C

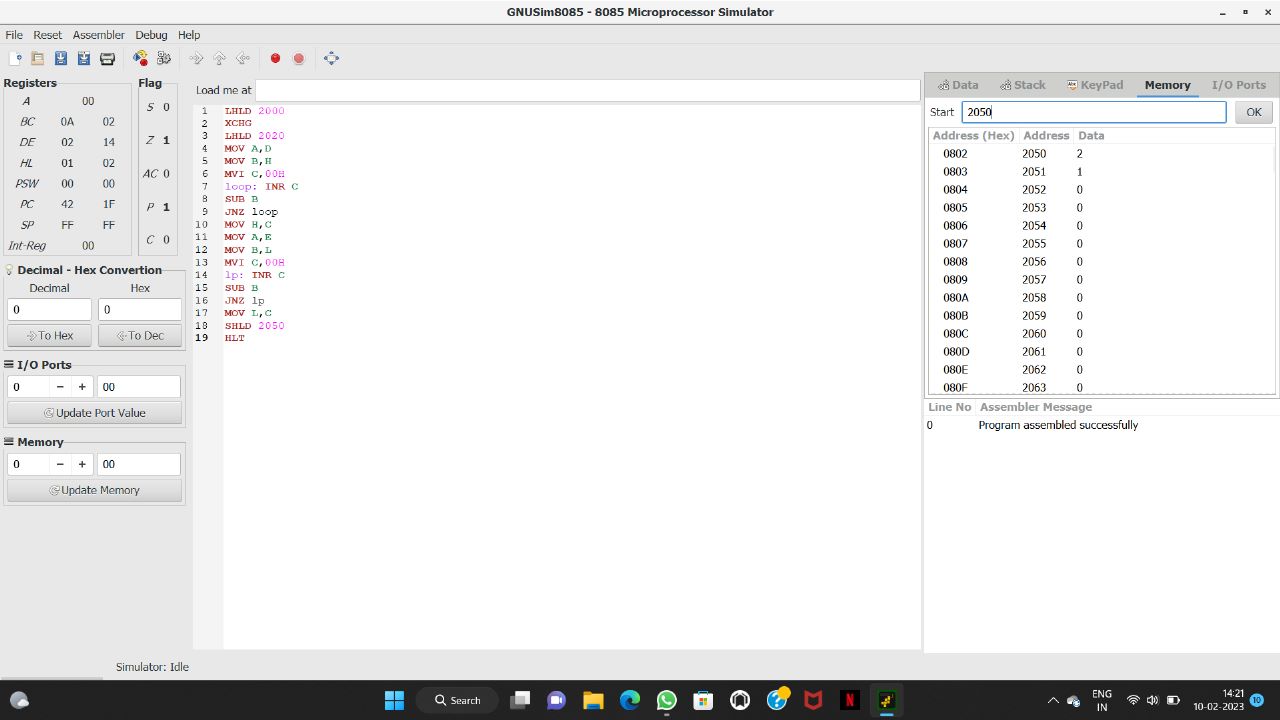
SUB B

JNZ lp

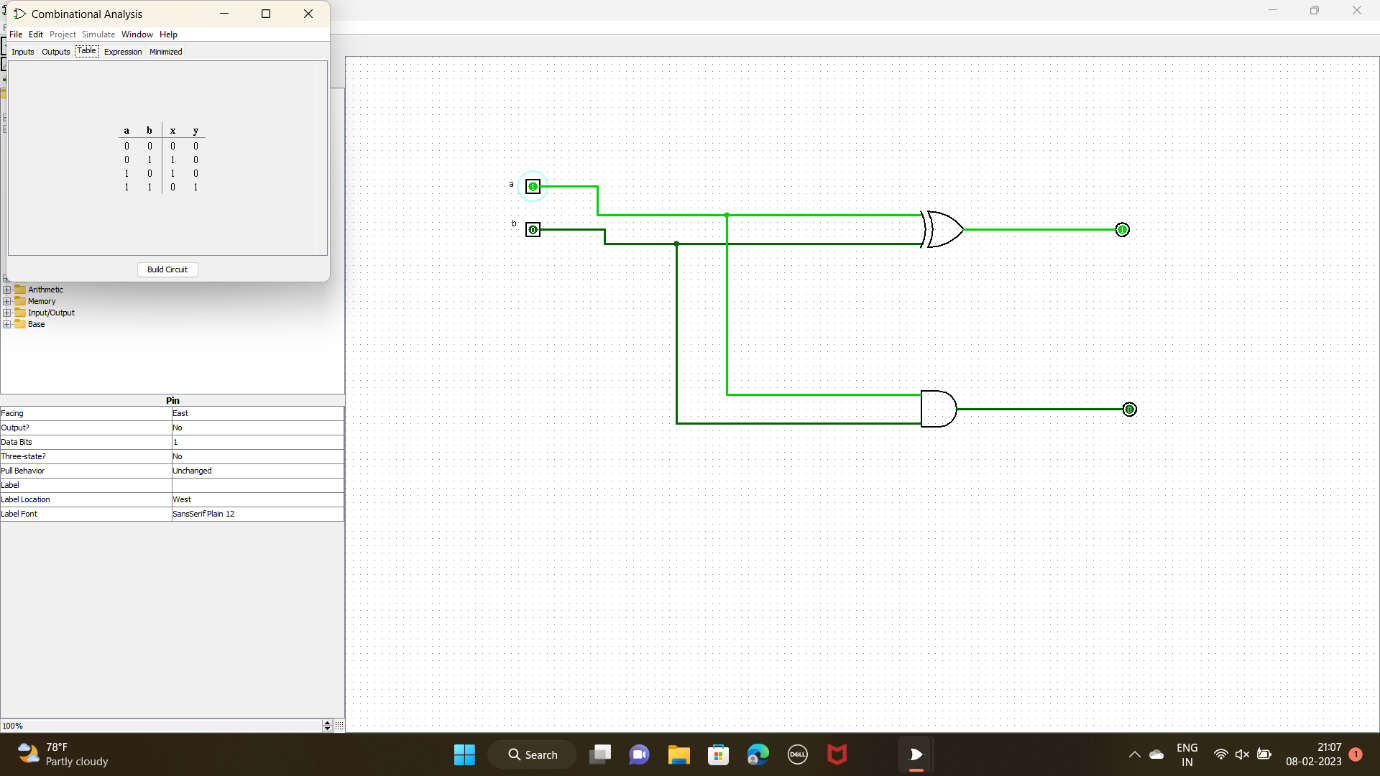
MOV L,C

SHLD 2050

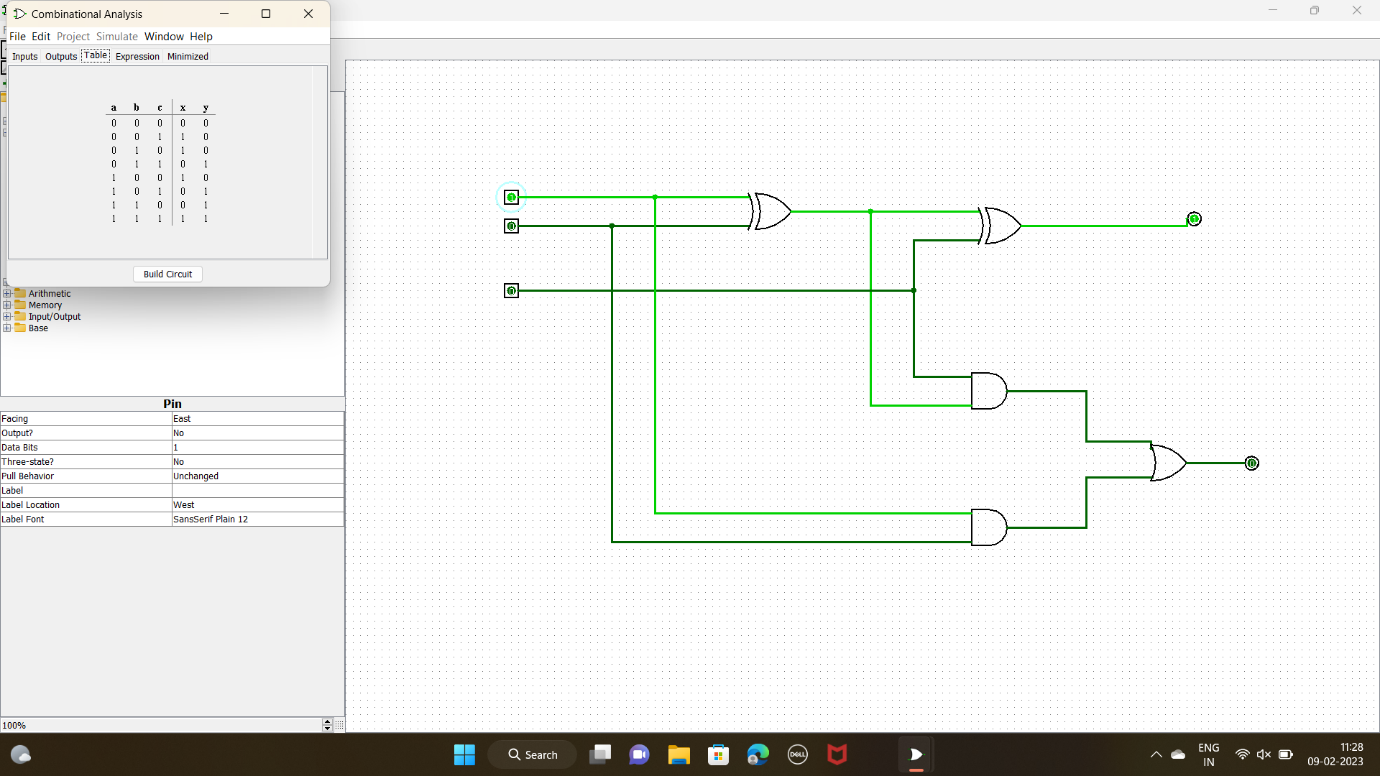
HLT



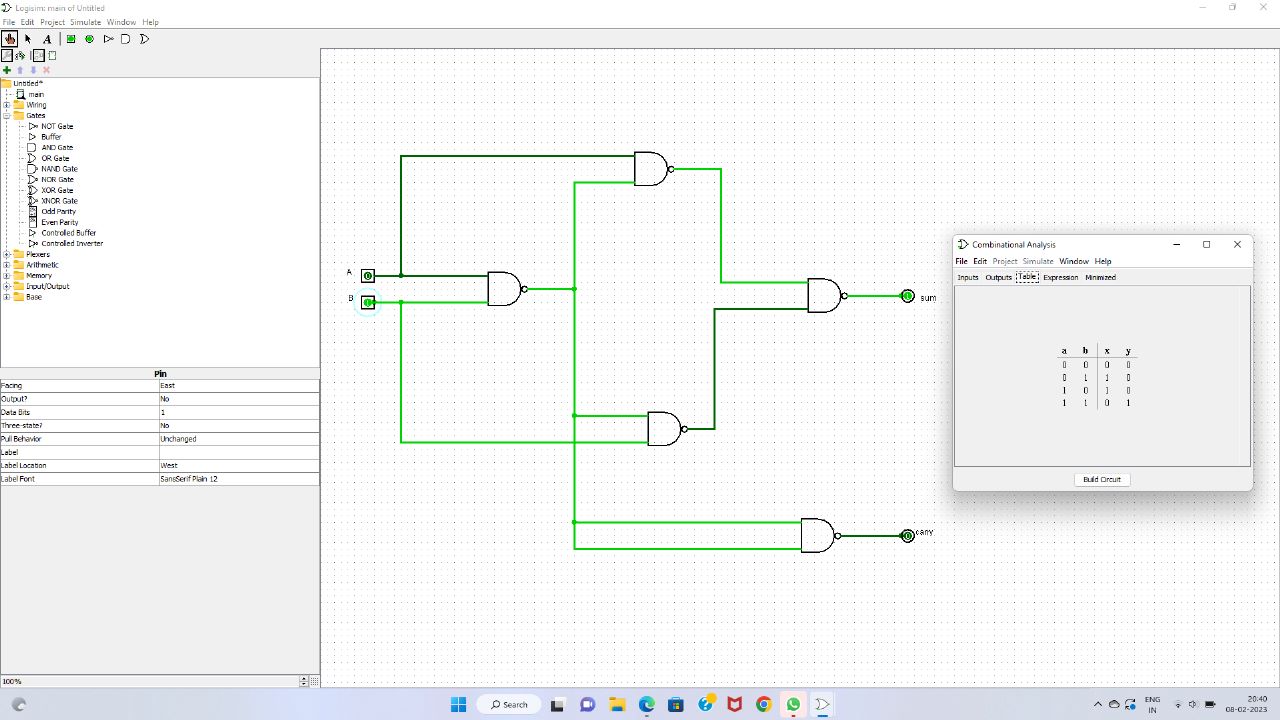
***9.2 BIT HALF ADDER***

******

***10.3 BIT FULL ADDER***

******

***11.2 BIT HALF ADDER WITH NAND GATES***



***12.FACTORIAL OF A NUMBER***

LXI H,2500H

MOV B,M

MVI D,01H

LOOP1: CALL FACT

DCR B

JNZ LOOP1

INX H

MOV M,D

HLT

FACT: MOV C,B

MVI A,00

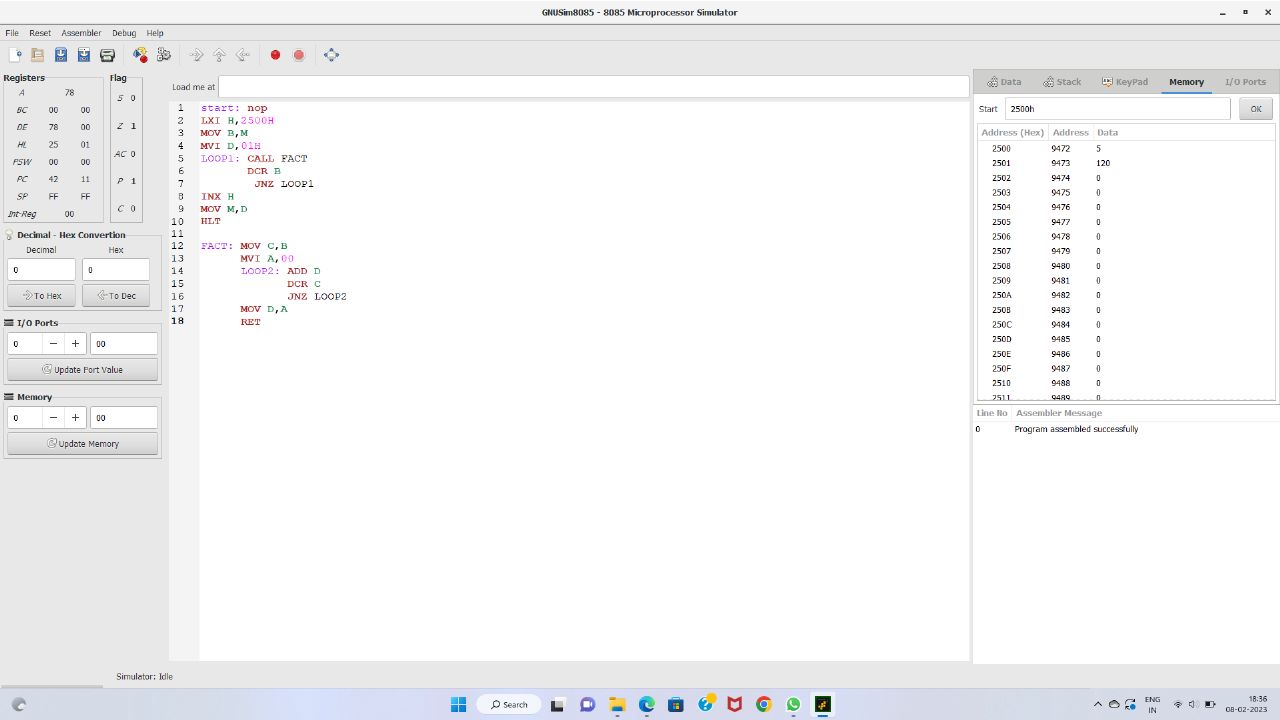
LOOP2: ADD D

DCR C

JNZ LOOP2

MOV D,A

      RET



***13.LARGEST NUMBER IN AN ARRAY***

LXI H,8000H

MOV C,M

INX H

MOV B,M

DCR C

LOOP: INX H

MOV A,M

CMP B

JC SKIP

MOV B,A

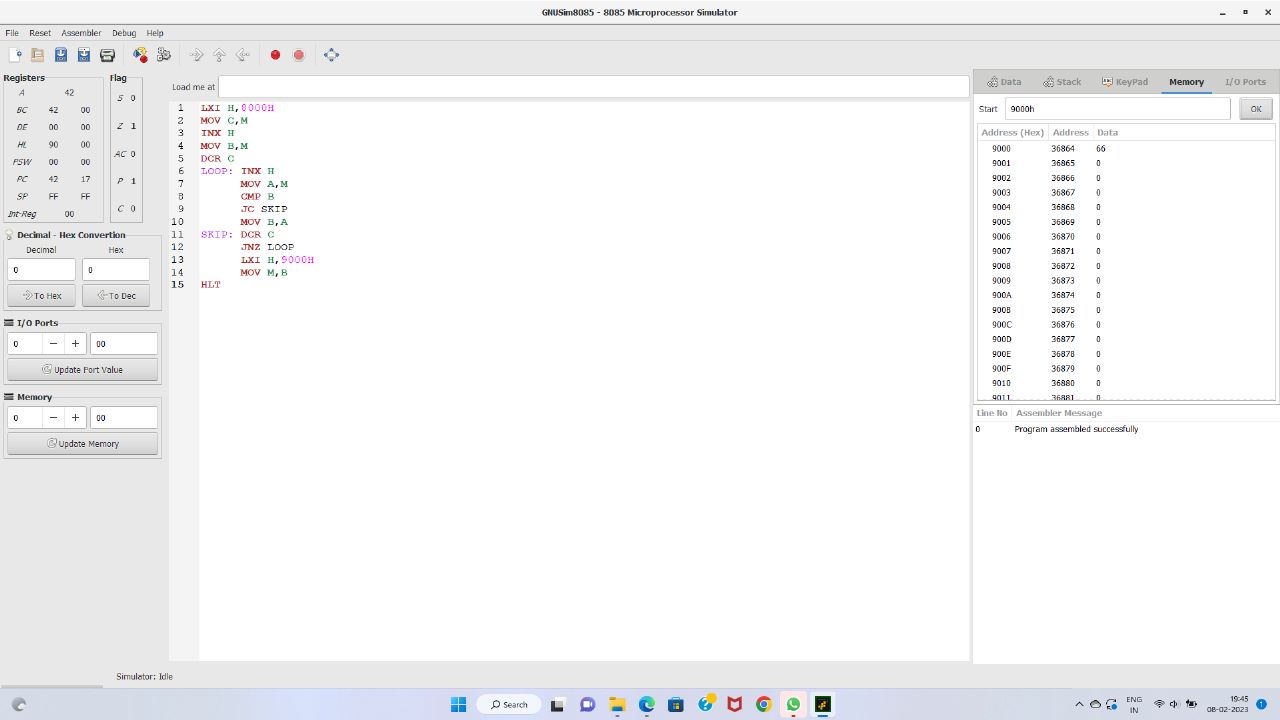
SKIP: DCR C

JNZ LOOP

LXI H,8008H

     MOV M,B

HLT



***14.2 STAGE PIPELINING***

#include<stdio.h>

int main()

{

int counter =1,a,b,choice,res;

float performance\_measure,ins;

printf("Enter number 1:");

scanf("%d",&a);

counter = counter+1;

printf("Enter number 2:");

scanf("%d",&b);

counter = counter +1;

printf("1-Addition:\n2-Subtraction:\n3-Multiplication:\n4-Division:");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("Performing addition\n");

res = a+b;

counter = counter+1;

break;

case 2: printf("Performing subtraction\n");

res = a-b;

counter = counter+1;

break;

case 3: printf("Performing Multiplication\n");

res = a\*b;

counter = counter+1;

break;

case 4: printf("Performing Division\n");

res = a/b;

counter = counter+1;

break;

default: printf("Wrong input");

break;

}

printf("The cycle value is:%d\n",counter);

printf("Enter the number of instructions:");

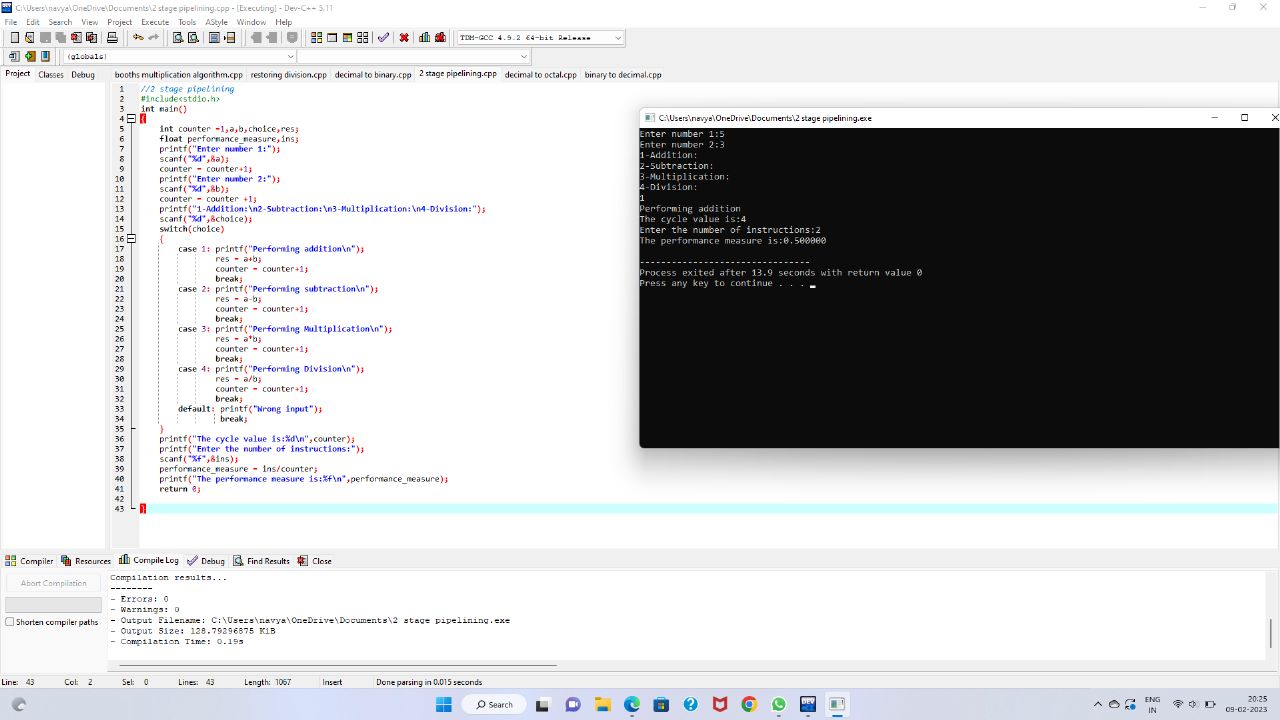
scanf("%f",&ins);

performance\_measure = ins/counter;

printf("The performance measure is:%f\n",performance\_measure);

return 0;

}



***15.3 STAGE PIPELINING***

#include<stdio.h>

int main()

{

float a,b,counter=1,res,INS;

float performance\_measure;

printf("Enter the number 1: ");

scanf("%f",&a);

printf("Enter the number 2: ");

scanf("%f",&b);

counter =counter+1;

res=a || b;

counter=counter+2;

printf("enter no.of instruction:");

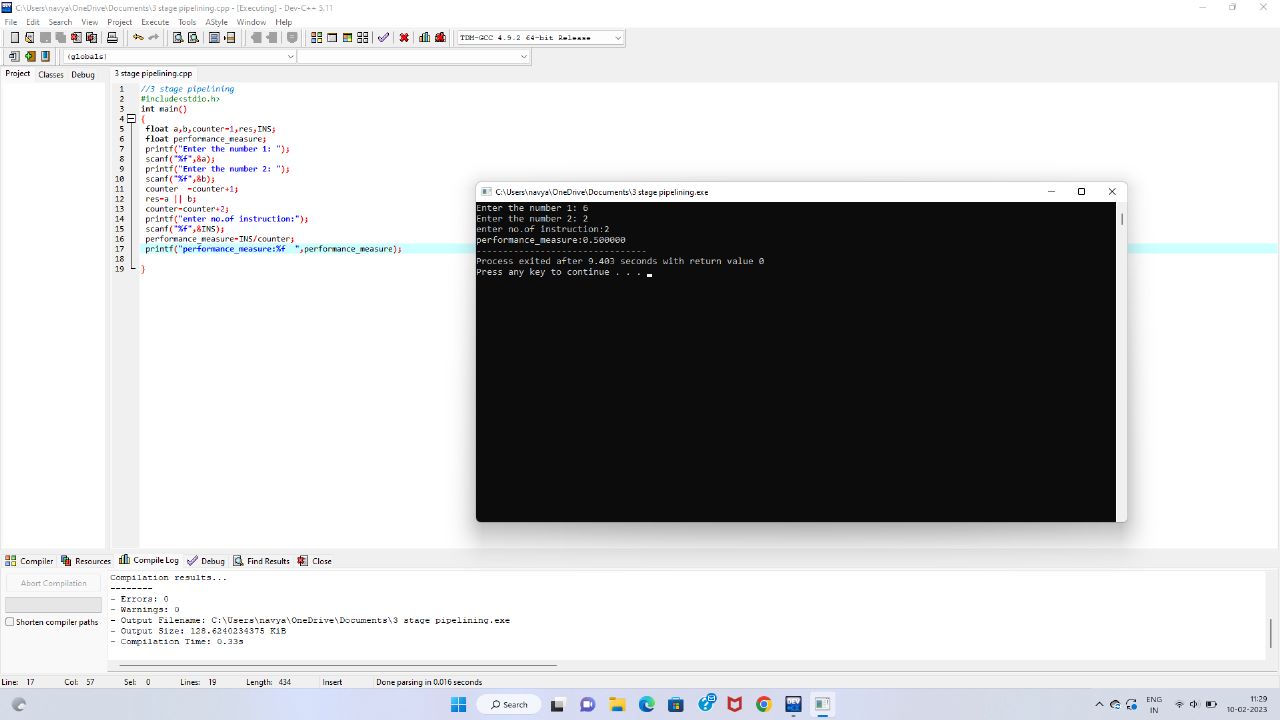
scanf("%f",&INS);

performance\_measure=INS/counter;

printf("performance\_measure:%f ",performance\_measure);

return 0;

}



***16.4 STAGE PIPELINING***

*#include<stdio.h>*

*int main(){*

*int counter=0;*

*int input;*

*int num1,num2;*

*int op;*

*int res;*

*int ins;*

*int performance\_measure=0;*

*printf("\n Enter 1st value: ");*

*scanf("%d",&num1);*

*counter+=1;*

*printf("\n Enter the 2nd value: ");*

*scanf("%d",&num2);*

*counter+=1;*

*printf("\n Enter the option: \n1)Addition\n2)Subraction\n3)Multiplication\n4)Division");*

*scanf("%d",&op);*

*switch(op){*

*case 1:*

*printf("Performing addition operation");*

*res=num1+num2;*

*counter+=1;*

*break;*

*case 2:*

*printf("Performing subraction operation");*

*res=num1-num2;*

*counter+=1;*

*break;*

*case 3:*

*printf("Performing multiplication operation");*

*res=num1\*num2;*

*counter+=1;*

*break;*

*case 4:*

*if(num2==0){*

*printf("\n Denominator can't be zero");*

*}*

*else{*

*printf("Performing division operation");*

*res=num1/num2;*

*counter+=1;*

*break;*

*}*

*default:*

*printf("Invalid case...");*

*counter+=3;*

*break;*

*}*

*printf("\n CYCLE VALUE IS : %d",counter);*

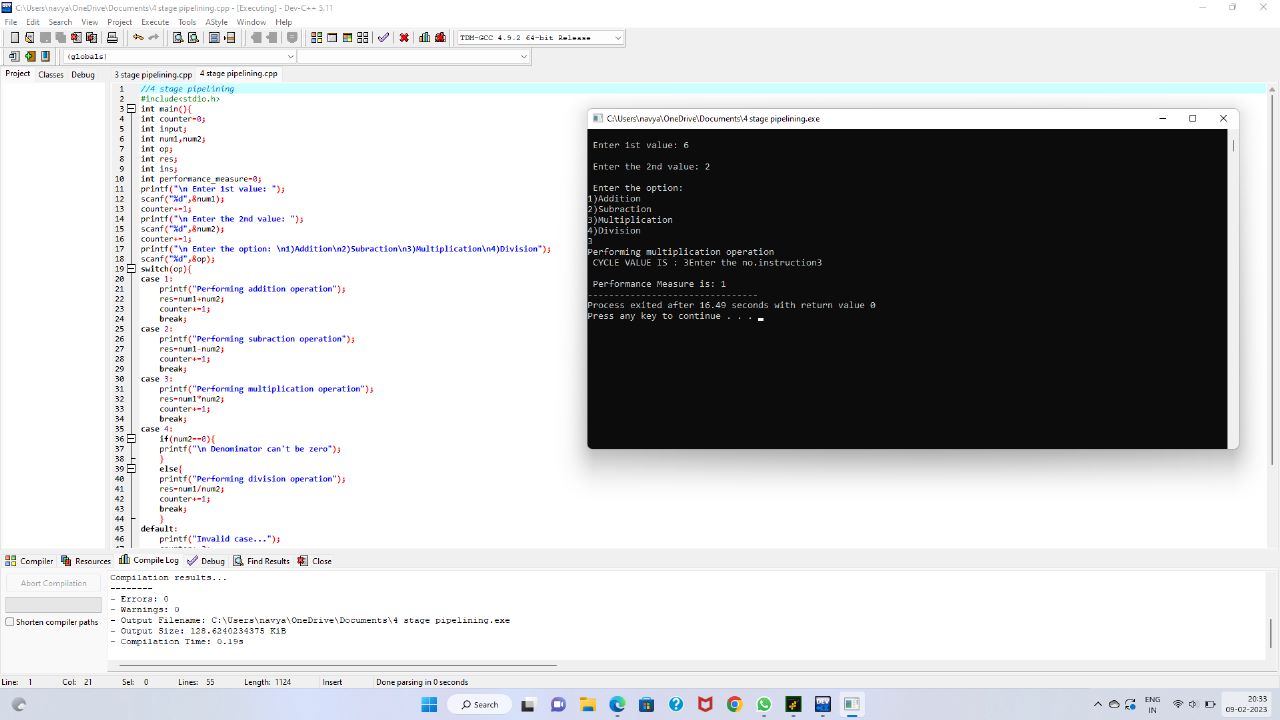
*printf("Enter the no.instruction");*

*scanf("%d",&ins);*

*performance\_measure=ins/counter;*

*printf("\n Performance Measure is: %d",performance\_measure);*

*}*



***17.BOOTHS MULTIPLICATION***

#include <stdio.h>

#include <math.h>

int a = 0,b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0};

int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};

int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0};

void binary(){

a1 = fabs(a);

b1 = fabs(b);

int r, r2, i, temp;

for (i = 0; i < 5; i++){

r = a1 % 2;

a1 = a1 / 2;

r2 = b1 % 2;

b1 = b1 / 2;

anum[i] = r;

anumcp[i] = r;

bnum[i] = r2;

if(r2 == 0){

bcomp[i] = 1;

}

if(r == 0){

acomp[i] =1;

}

}

c = 0;

for ( i = 0; i < 5; i++){

res[i] = com[i]+ bcomp[i] + c;

if(res[i] >= 2){

c = 1;

}

else

c = 0;

res[i] = res[i] % 2;

}

for (i = 4; i >= 0; i--){

bcomp[i] = res[i];

}

if (a < 0){

c = 0;

for (i = 4; i >= 0; i--){

res[i] = 0;

}

for ( i = 0; i < 5; i++){

res[i] = com[i] + acomp[i] + c;

if (res[i] >= 2){

c = 1;

}

else

c = 0;

res[i] = res[i]%2;

}

for (i = 4; i >= 0; i--){

anum[i] = res[i];

anumcp[i] = res[i];

}

}

if(b < 0){

for (i = 0; i < 5; i++){

temp = bnum[i];

bnum[i] = bcomp[i];

bcomp[i] = temp;

}

}

}

void add(int num[]){

int i;

c = 0;

for ( i = 0; i < 5; i++){

res[i] = pro[i] + num[i] + c;

if (res[i] >= 2){

c = 1;

}

else{

c = 0;

}

res[i] = res[i]%2;

}

for (i = 4; i >= 0; i--){

pro[i] = res[i];

printf("%d",pro[i]);

}

printf(":");

for (i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}

void arshift(){

int temp = pro[4], temp2 = pro[0], i;

for (i = 1; i < 5 ; i++){

pro[i-1] = pro[i];

}

pro[4] = temp;

for (i = 1; i < 5 ; i++){

anumcp[i-1] = anumcp[i];

}

anumcp[4] = temp2;

printf("\nAR-SHIFT: ");

for (i = 4; i >= 0; i--){

printf("%d",pro[i]);

}

printf(":");

for(i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}

int main(){

int i, q = 0;

printf("\t\tBOOTH'S MULTIPLICATION ALGORITHM");

printf("\nEnter two numbers to multiply: ");

printf("\nBoth must be less than 16");

do{

printf("\nEnter A: ");

scanf("%d",&a);

printf("Enter B: ");

scanf("%d", &b);

}while(a >=16 || b >=16);

printf("\nExpected product = %d", a \* b);

binary();

printf("\n\nBinary Equivalents are: ");

printf("\nA = ");

for (i = 4; i >= 0; i--){

printf("%d", anum[i]);

}

printf("\nB = ");

for (i = 4; i >= 0; i--){

printf("%d", bnum[i]);

}

printf("\nB'+ 1 = ");

for (i = 4; i >= 0; i--){

printf("%d", bcomp[i]);

}

printf("\n\n");

for (i = 0;i < 5; i++){

if (anum[i] == q){

printf("\n-->");

arshift();

q = anum[i];

}

else if(anum[i] == 1 && q == 0){

printf("\n-->");

printf("\nSUB B: ");

add(bcomp);

arshift();

q = anum[i];

}

else{

printf("\n-->");

printf("\nADD B: ");

add(bnum);

arshift();

q = anum[i];

}

}

printf("\nProduct is = ");

for (i = 4; i >= 0; i--){

printf("%d", pro[i]);

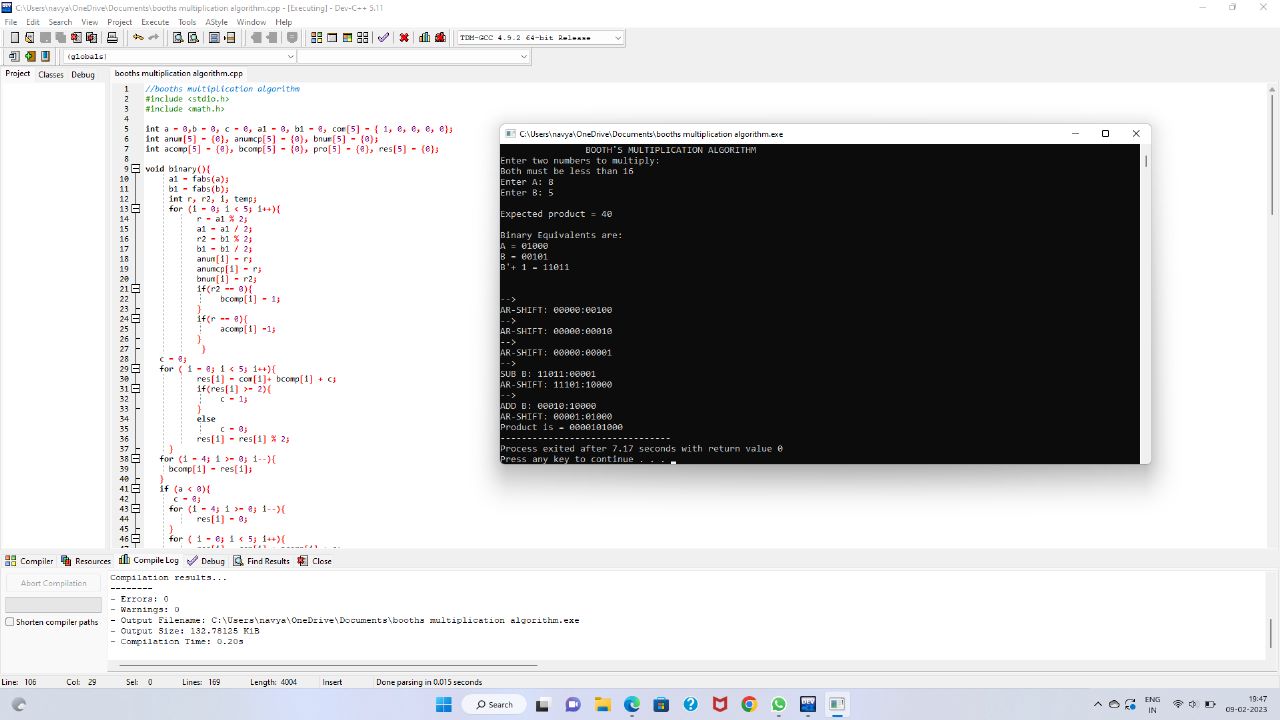
}

for (i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}



***18.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)

{

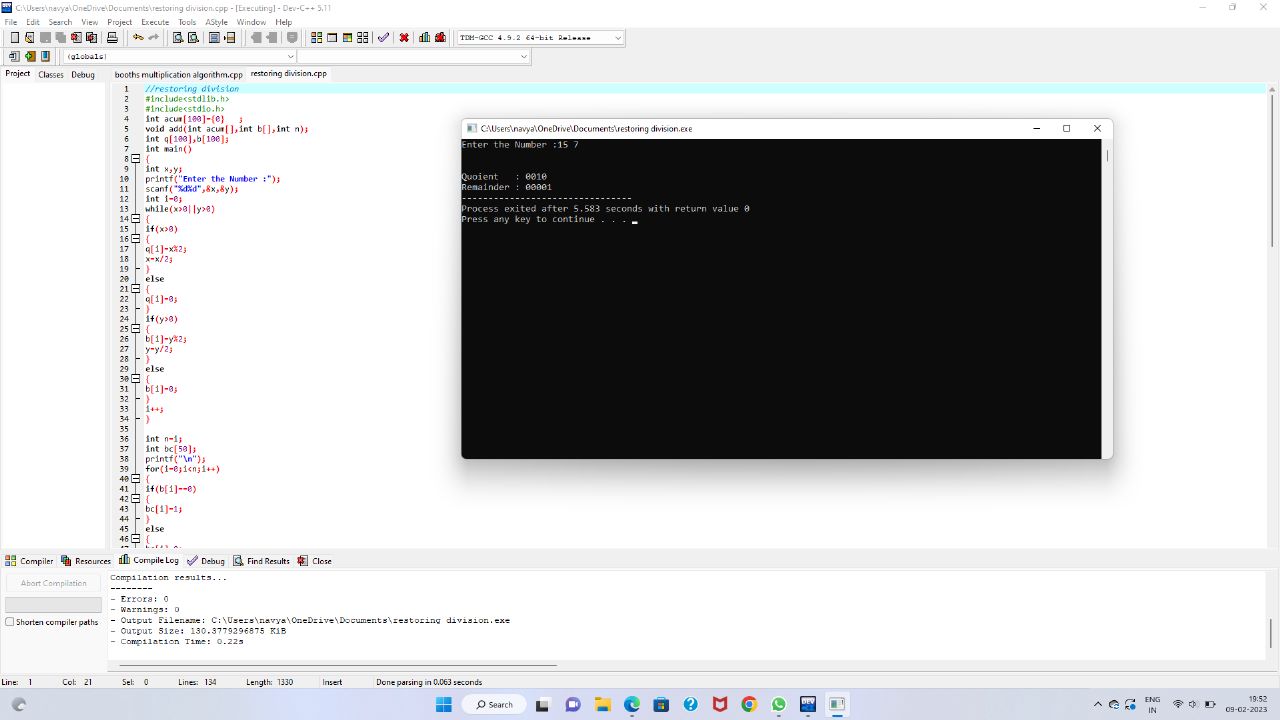
acum[i]=1;

temp=1;

}

}

}



***19.Cache hit ratio***

#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:");

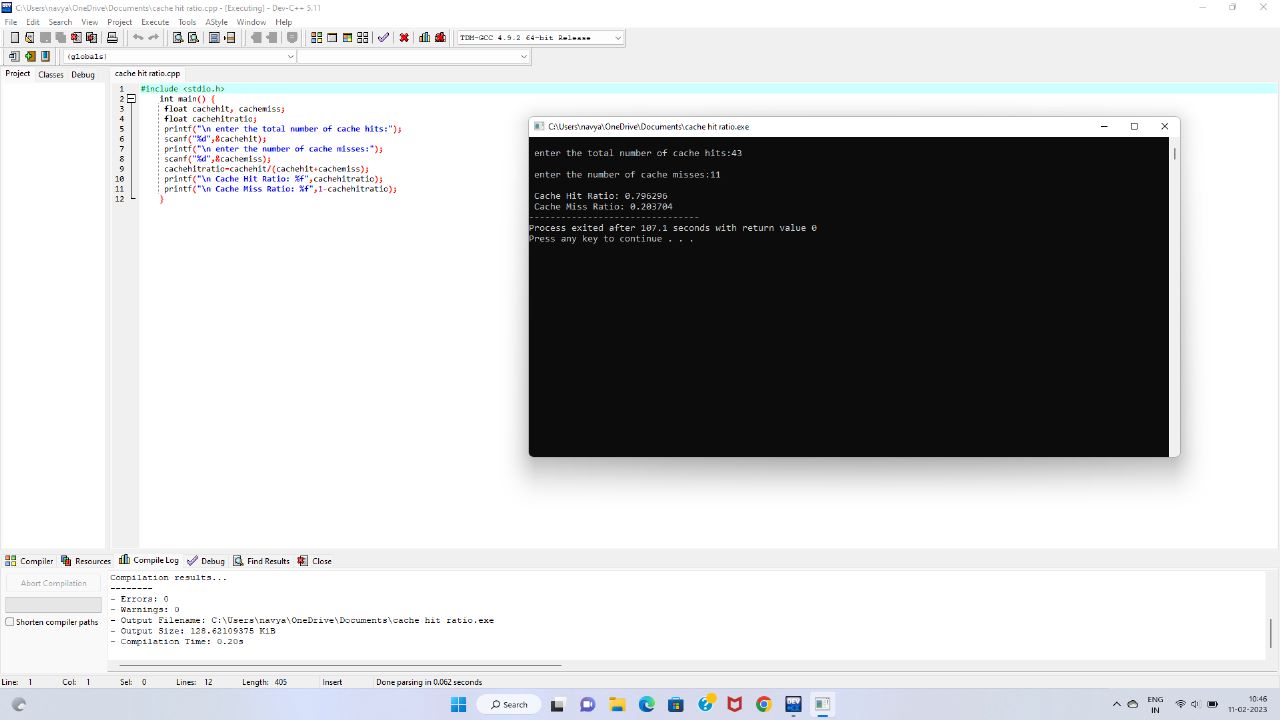
scanf("%d",&cachemiss);

cachehitratio=cachehit/(cachehit+cachemiss);

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

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

}



***20.ONES AND TWOS COMPLIMENT***

LDA 300H

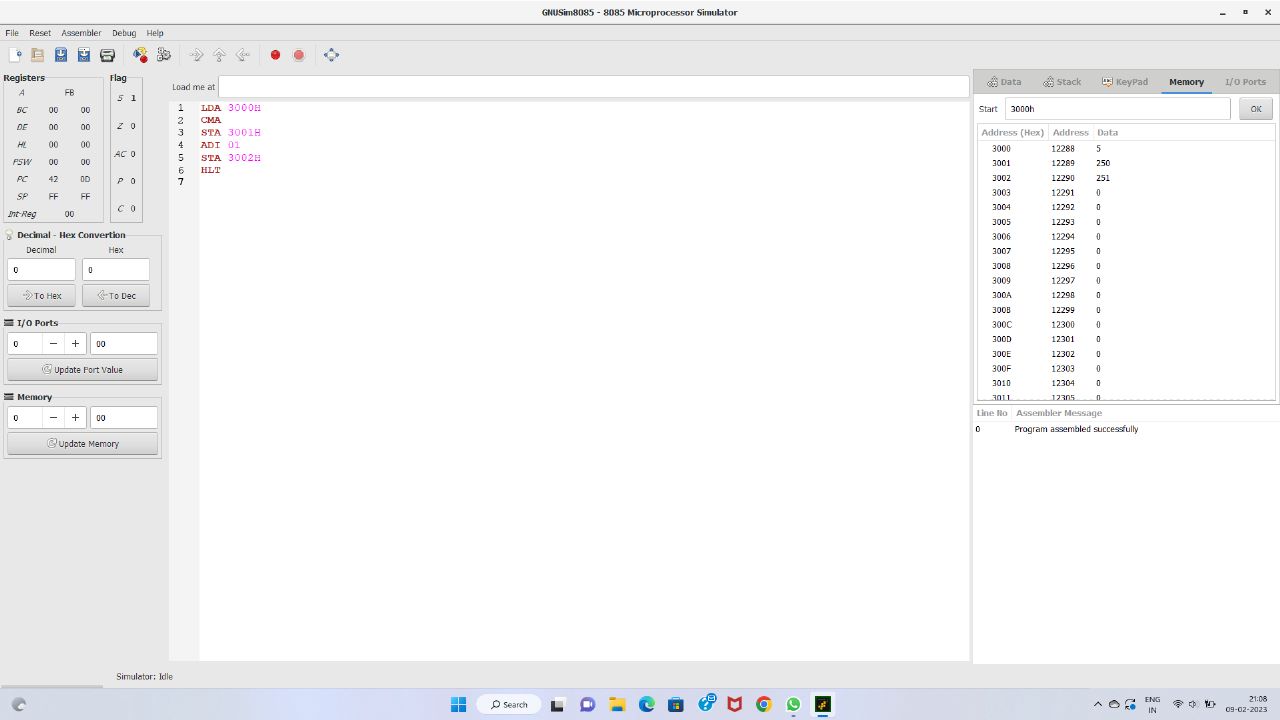
CMA

STA 3001

ADI 01

STA 3002

HLT



***21.DECIMAL TO BINARY***

#include <stdio.h>

#include <math.h>

int convert(long long);

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;

}

int convert(long long n) {

int dec = 0, i = 0, rem;

while (n!=0) {

rem = n % 10;

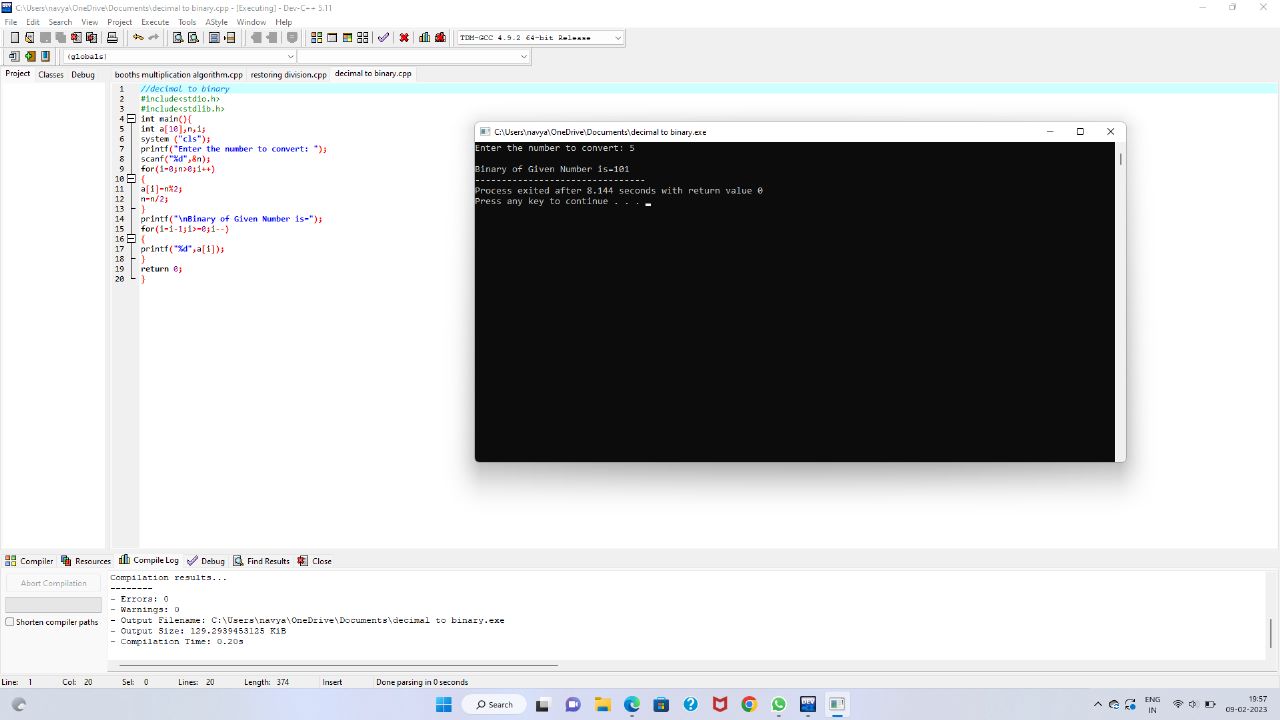
n /= 10;

dec += rem \* pow(2, i);

++i;

}

}



***22.DECIMAL TO OCTAL***

#include <stdio.h>

int main()

{

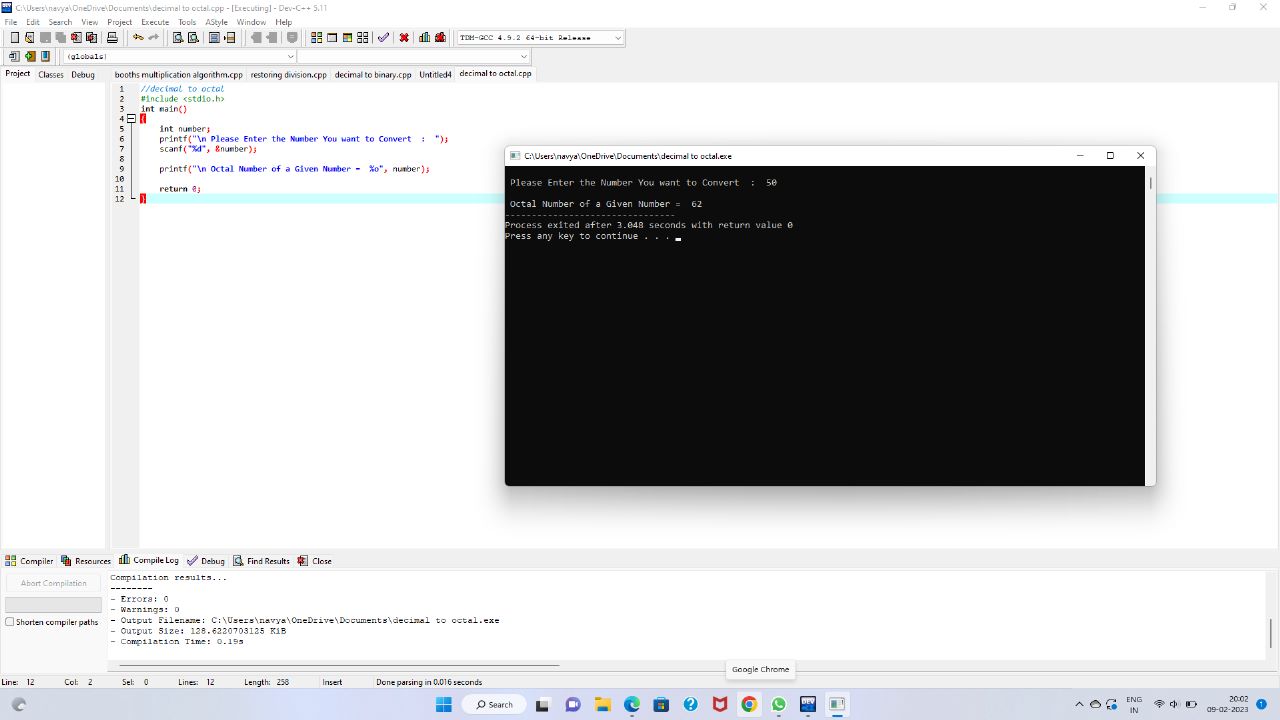
int number;

printf("\n Please Enter the Number You want to Convert : ");

scanf("%d", &number);

printf("\n Octal Number of a Given Number = %o", number);

}



***23.BINARY TO DECIMAL***

#include <stdio.h>

#include <math.h>

int convert(long long);

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;

}

int convert(long long n) {

int dec = 0, i = 0, rem;

while (n!=0) {

rem = n % 10;

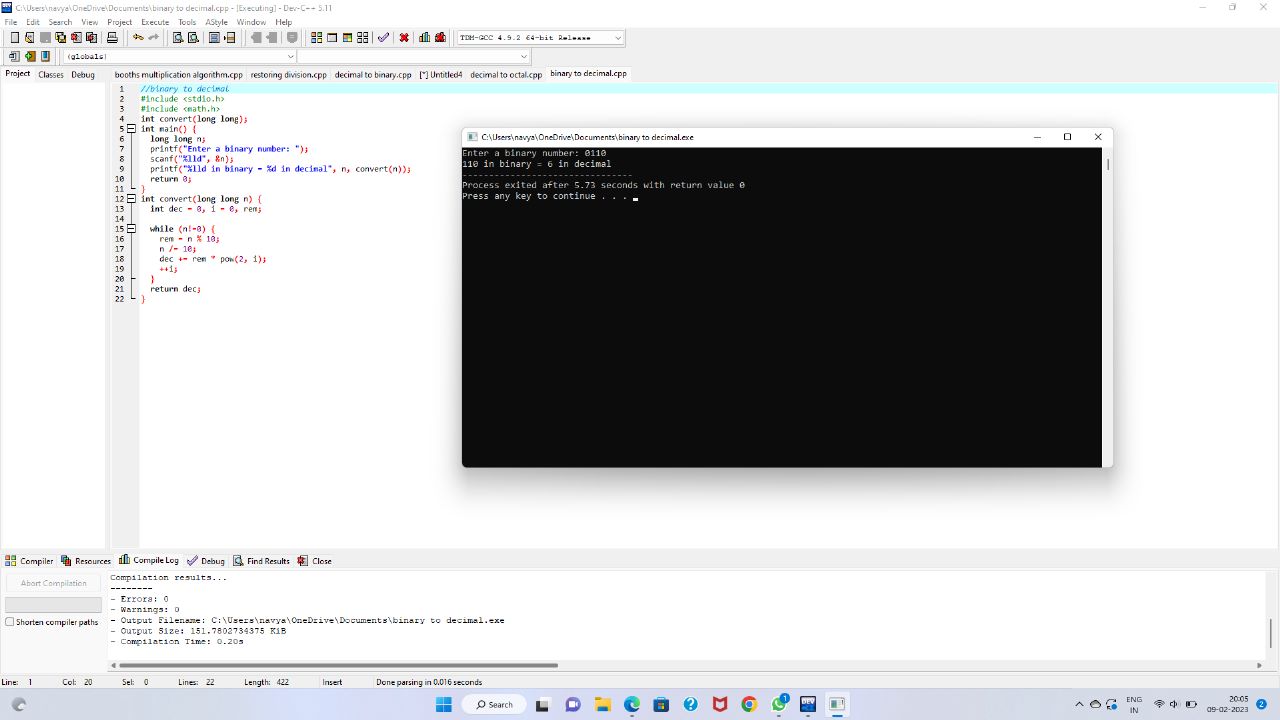
n /= 10;

dec += rem \* pow(2, i);

++i;

}

}



***24.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++)

{

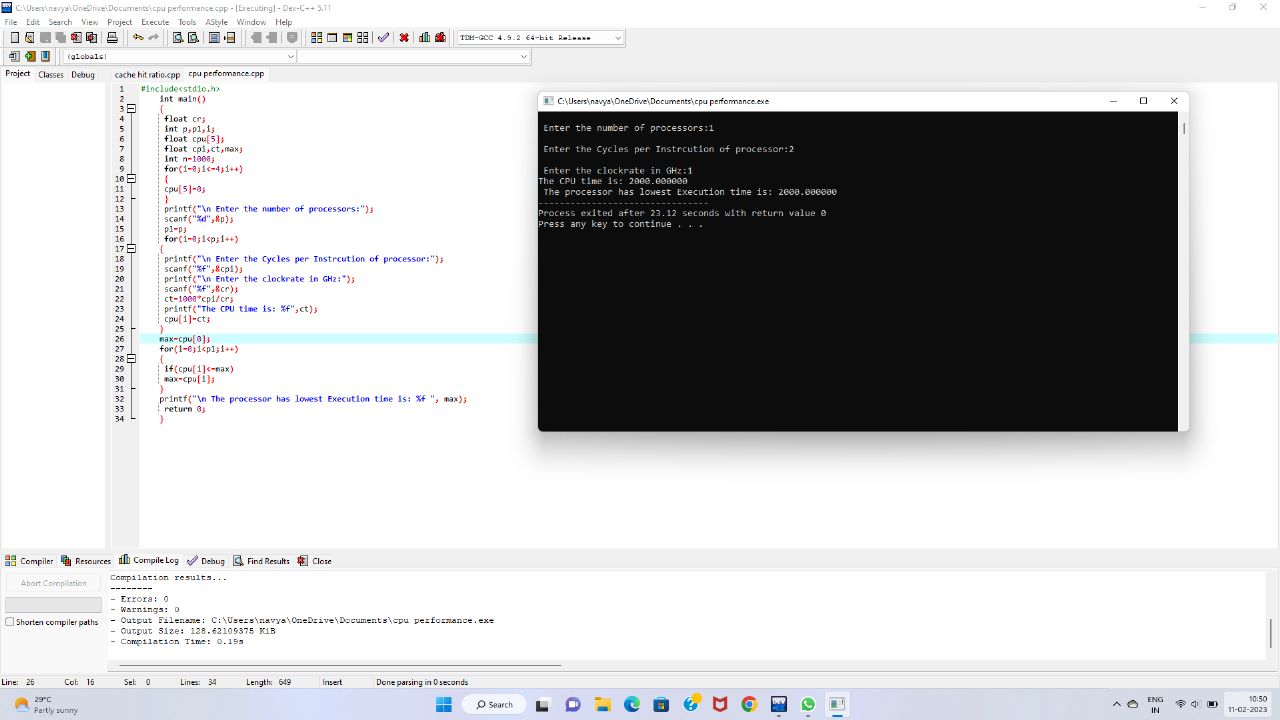
if(cpu[i]<=max)

max=cpu[i];

}

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

}



***25.SWAP TWO NUMBERS***

LDA 2500

MOV B,A

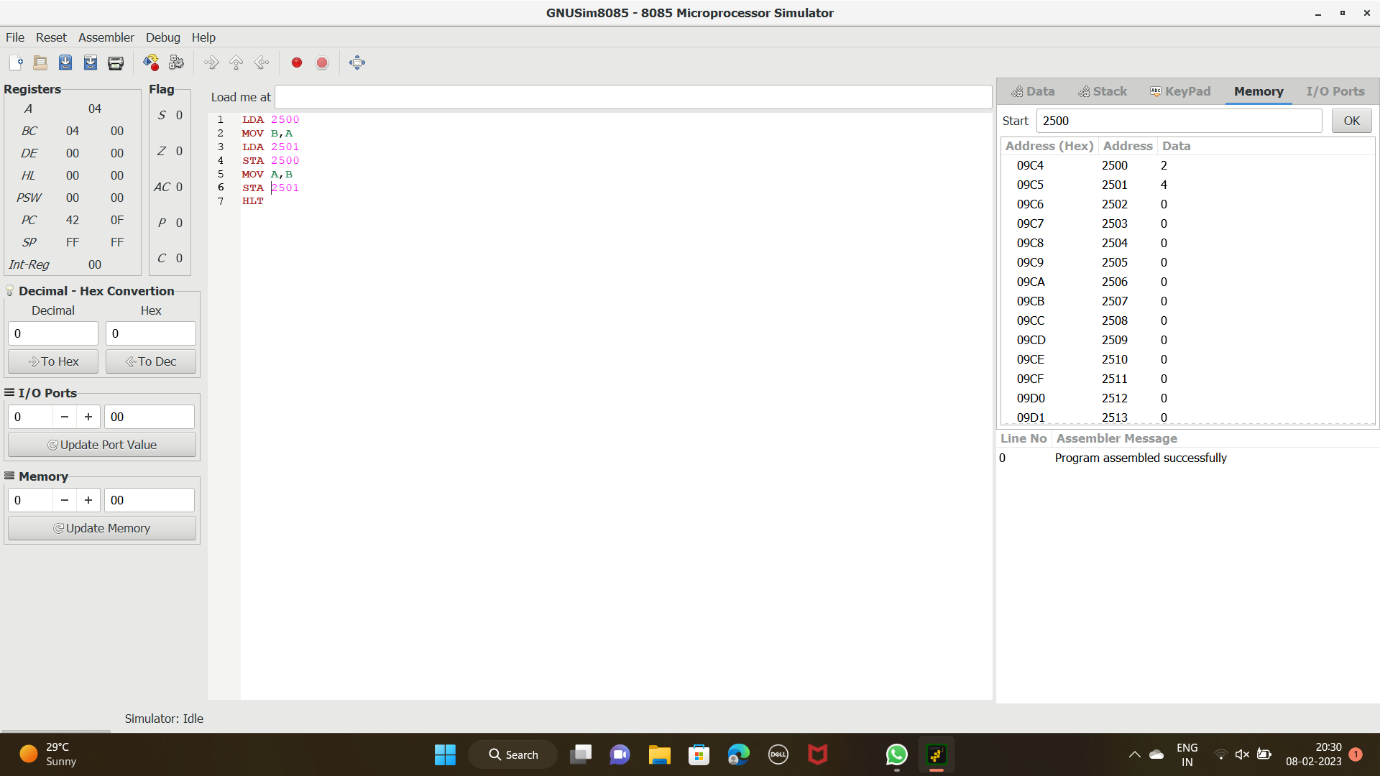
LDA 2501

STA 2500

MOV A,B

STA 2501

HLT

******