EXPERMENT 1:

|  |
| --- |
|  |
|  | LDA 8500  MOV B,A |
|  | LDA 8501 |
|  | ADD B |
|  | STA 8502 |
|  | RST 1 |

EXPERIMENT 2:

|  |
| --- |
|  |
|  | LDA 8500  MOV B,A |
|  | LDA 8501 |
|  | SUB B |
|  | STA 8502 |
|  | RST 1 |

EXPERMENT 3:

|  |
| --- |
|  |
|  | LDA 8500  MOV B,A |
|  | LDA 8501 |
|  | MOV C,A |
|  | LDA 8502 |
|  | MOV D,A |
|  | LDA 8503 |
|  | ADD C |
|  | STA 8505 |
|  | MOV A,D |
|  | ADD B |
|  | STA 8504 |
|  | RST 1 |

EXPERIMENT 4:

|  |
| --- |
|  |
|  | LHLD 2500  XCHG |
|  | LHLD 2501 |
|  | MOV A,E |
|  | SUB L |
|  | MOV L,A |
|  | MOV A,D |
|  | ADC H |
|  | MOV H,A |
|  | SHLD 2502 |
|  | HLT |

EXPERMENT 5:

|  |
| --- |
|  |
|  | 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 |

EXPERMENT 6:

|  |
| --- |
|  |
|  | LHLD 8500  MOV D,H |
|  | MOV E,L |
|  | LDA 8502 |
|  | MOV C,A |
|  | CPI 00 |
|  | JZ LOOP1 |
|  | LXI H,0000 |
|  | LOOP: DAD D |
|  | DCR C |
|  | JZ LOOP1 |
|  | LOOP1: SHLD 8503 |
|  | HLT |

EXPERIMENT 7:

|  |
| --- |
|  |
|  | LHLD 8500  MOV D,H | |
|  | MOV E,L | |
|  | LDA 8502 | |
|  | MOV C,A | |
|  | CPI 00 | |
|  | JZ LOOP1 | |
|  | LXI H,0000 | |
|  | LOOP: DAD D | |
|  | DCR C | |
|  | JZ LOOP1 | |
|  | LOOP1: SHLD 8503 | |
|  | HLT | |
| EXP 13 |  | |
|  |  | |
|  | | LXI H,8000  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,8010 |
|  | | MOV M,B |
|  | | HLT |

EXPERIMENT 14:

|  |  |
| --- | --- |
|  | #include<stdio.h>  int main() |
|  | { |
|  | int counter =1,a,b,choice,res,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("%d",&ins); |
|  | int performance\_measure = ins/counter; |
|  | printf("The performance measure is:%d\n",performance\_measure); |
|  | return 0; |
|  | } |

EXPERIMENT 15:

|  |
| --- |
| #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; |

EXPERIMENT 16:

#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;

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;

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]);

}

}

void 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]);

}

}

EXPERIMENT 18:

#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;

}

}

}

EXPERIMENT 20:

LDA 8000

CMA

STA 8010

INR A

STA 8011

HLT

EXPERIMENT 21:

1. #include<stdio.h>

2. #include<stdlib.h>

3. int main(){

4. int a[10],n,i;

5. system ("cls");

6. printf("Enter the number to convert: ");

7. scanf("%d",&n);

8. for(i=0;n>0;i++)

9. {

10. a[i]=n%2;

11. n=n/2;

12. }

13. printf("\nBinary of Given Number is=");

14. for(i=i-1;i>=0;i--)

15. {

16. printf("%d",a[i]);

17. }

18. return 0;

19. }

EXPERIMENT 22:

#include <stdio.h>

int main()

{

long decimalnum, remainder, quotient,octalnum=0;

int octalNumber[100], i = 1, j;

printf("Enter the decimal number: ");

scanf("%ld", &decimalnum);

quotient = decimalnum;

while (quotient != 0)

{

octalNumber[i++] = quotient % 8;

quotient = quotient / 8;

}

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

octalnum = octalnum\*10 + octalNumber[j];

printf("Equivalent octal value of decimal no %d is: %d

return 0;

}

EXPERIMENT 23:

#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;

}

return dec;

}

EXPERIMENT:24

#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++)

{