

# Contents

<b>Sr. No.</b>	<b>Title of the Practical:</b>	<b>Page no</b>	<b>Signature</b>
01.	To Design and train a perceptron for AND Gate.		
02.	To design and train a perceptron training for OR gate.		
03.	To design and train a perceptron training for EX-OR gate.		
04.	To design and train a perceptron for NOT gate.		
05.	To design and train a perceptron for identifying ODD and EVEN number.		
06.	To create a Bi-directional Associative Memory (BAM) for ID and telephone number.		
07.	To design and train the Hopfield net to map the input vector with the stored vector and correct them.		

## PRACTICAL NO – 1

**Aim: To Design and train a perceptron for AND Gate.**

Program:

```
#include<stdio.h>
void main()
{
int x0=1,d=1,w0=0,w1=0,w2=0,i,j,yin,net;
int x1[4];//={ 1,1,-1,-1 };
int x2[4];//={ 1,-1,1,-1 };
int t[4];//={ 1,-1,-1,-1 };
int x,y=4;
char b[5];
i=0;
clrscr();
printf("Enter the Name of The Gate=>");
gets(b);
printf("Enter The Truth Table For the '%s' Gate\n",b);
printf("x1\tx2\tt\n");
for(j=0;j<4;j++)
{
scanf("%d",&x1[j]);
x=9;
gotoxy(x,y);
scanf("%d",&x2[j]);
gotoxy(x+8,y);
scanf("%d",&t[j]);
x=0;y++;
}
for(j=0;j<4;j++)
{ if(x1[j]==0) x1[j]=-1;
if(x2[j]==0) x2[j]=-1;
if(t[j]==0) t[j]=-1;
}
yin= w0+((x1[0]*w1)+(x2[0]*w2));
yin=yin>0?1:(yin==0?0:-1);
printf("\nyin=%d\n",yin);
while((yin!=t[i])&&(i<4))
{
printf(" \nt=%d",t[i]);
w0=w0+d*t[i]*x0;
printf("\nw0=w0+d*t*x0=%d",w0);
w1=w1+d*t[i]*x1[i];
printf("\nw1=w1+d*t*x1=%d",w1);
w2=w2+d*t[i]*x2[i];
printf("\nw2=w2+d*t*x2=%d",w2);
printf("\nNew Matrix of weights is { %d %d %d }",w0,w1,w2);
i++;
yin= w0+((x1[i]*w1)+(x2[i]*w2));
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Name of The Gate=>AND
Enter The Truth Table For the 'AND' Gate
x1      x2      t
0       0       0
0       1       0
1       0       0
1       1       1

yin=0

t=-1
w0=w0+d*t*x0=-1
w1=w1+d*t*x1=1
w2=w2+d*t*x2=1
New Matrix of weights is <-1 1 1>
yin=-1 t=-1
x1      x2      NET
0       0       0
0       1       0
1       0       0
1       1       1
Perceptron is Trained sucessfully for 'AND' Gate

-----
Process exited with return value 0
Press any key to continue . . .
```

```

yin=yin>0?1:(yin==0?0:-1);
printf("\nyin=%d",yin);
}
printf(" t=%d",t[i]);
if(yin!=t[i])
printf("\nPerceptron can't be trained for '%s' gate" ,b);
else{
printf("\nx1\tx2\tNET");
for(i=0;i<=3;i++)
{
net= (x0*w0)+(x1[i]*w1)+(x2[i]*w2);
net=net<0?0:1;
if(x1[i]==-1)
x1[i]=0;
if(x2[i]==-1)
x2[i]=0;
printf("\n%d\t%d\t%d",x1[i],x2[i],net);
}
printf("\nPerceptron is Trained sucessfully for '%s' Gate\n",b);
}
getch();
}

```



## PROGRAM NO- 2

**Aim: To design and train perceptron training for OR gate.**

Program:

```
#include<stdio.h>
void main()
{
int x0=1,d=1,w0=0,w1=0,w2=0,i,j,yin,net;
int x1[4];//={ 1,1,-1,-1 };
int x2[4];//={ 1,-1,1,-1 };
int t[4];//={ 1,-1,-1,-1 };
int x,y=4;
char b[5];
i=0;
clrscr();
printf("Enter the Name of The Gate=>");
gets(b);
printf("Enter The Truth Table For the '%s' Gate\n",b);
printf("x1\tx2\tt\n");
for(j=0;j<4;j++)
{
scanf("%d",&x1[j]);
x=9;
gotoxy(x,y);
scanf("%d",&x2[j]);
gotoxy(x+8,y);
scanf("%d",&t[j]);
x=0;y++;
}
for(j=0;j<4;j++)
{ if(x1[j]==0) x1[j]=-1;
if(x2[j]==0) x2[j]=-1;
if(t[j]==0) t[j]=-1;
}
yin= w0+((x1[0]*w1)+(x2[0]*w2));
yin=yin>0?1:(yin==0?0:-1);
printf("yin=%d\n",yin);
while((yin!=t[i])&&(i<4))
{
printf(" t=%d",t[i]);
w0=w0+d*t[i]*x0;
printf("\nw0=w0+d*t*x0=%d",w0);
w1=w1+d*t[i]*x1[i];
printf("\nw1=w1+d*t*x1=%d",w1);
w2=w2+d*t[i]*x2[i];
printf("\nw2=w2+d*t*x2=%d",w2);
printf("\nNew Matrix of weights is { %d %d %d }",w0,w1,w2);
i++;
yin= w0+((x1[i]*w1)+(x2[i]*w2));
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Name of The Gate=>OR
Enter The Truth Table For the 'OR' Gate
x1    x2    t
0     0     0
1     0     1
0     1     1
1     1     1

yin=0
t=-1
    w0=w0+d*t*x0=-1
    w1=w1+d*t*x1=1
    w2=w2+d*t*x2=1
New Matrix of weights is <-1 1 1>
yin=-1
t=1
    w0=w0+d*t*x0=0
    w1=w1+d*t*x1=2
    w2=w2+d*t*x2=0
New Matrix of weights is <0 2 0>
yin=-1
t=1
    w0=w0+d*t*x0=1
    w1=w1+d*t*x1=1
    w2=w2+d*t*x2=1
New Matrix of weights is <1 1 1>
yin=1 t=1
x1    x2    NET
0     0     0
1     0     1
0     1     1
1     1     1
Perceptron is Trained sucessfully for 'OR' Gate

-----
Process exited with return value 0
Press any key to continue . . . _
```

```

yin=yin>0?1:(yin==0?0:-1);
printf("\nyin=%d",yin);
}
printf(" t=%d",t[i]);
if(yin!=t[i])
printf("\nPerceptron can't be trained for '%s' gate" ,b);
else{
printf("\nx1\tx2\tNET");
for(i=0;i<=3;i++)
{
net= (x0*w0)+(x1[i]*w1)+(x2[i]*w2);
net=net<0?0:1;
if(x1[i]==-1)
x1[i]=0;
if(x2[i]==-1)
x2[i]=0;
printf("\n%d\t%d\t%d",x1[i],x2[i],net);
}
printf("\nPerceptron is Trained sucessfully for '%s' Gate\n",b);
}
getch();
}

```





## PRACTICAL NO – 3

**Aim: To design and train a perceptron training for EX-OR gate.**

Program:

```
#include<stdio.h>
void main()
{
int x0=1,d=1,w0=0,w1=0,w2=0,i,j,yin,net;
int x1[4];//={ 1,1,-1,-1 };
int x2[4];//={ 1,-1,1,-1 };
int t[4];//={ 1,-1,-1,-1 };
int x,y=4;
char b[5];
i=0;
clrscr();
printf("Enter the Name of The Gate=>");
gets(b);
printf("Enter The Truth Table For the '%s' Gate\n",b);
printf("x1\tx2\tt\n");
for(j=0;j<4;j++)
{
scanf("%d",&x1[j]);
x=9;
gotoxy(x,y);
scanf("%d",&x2[j]);
gotoxy(x+8,y);
scanf("%d",&t[j]);
x=0;y++;
}
for(j=0;j<4;j++)
{ if(x1[j]==0) x1[j]=-1;
if(x2[j]==0) x2[j]=-1;
if(t[j]==0) t[j]=-1;
}
yin= w0+((x1[0]*w1)+(x2[0]*w2));
yin=yin>0?1:(yin==0?0:-1);
printf("yin=%d\n",yin);
while((yin!=t[i])&&(i<4))
{
printf(" t=%d",t[i]);
w0=w0+d*t[i]*x0;
printf("\nw0=w0+d*t*x0=%d",w0);
w1=w1+d*t[i]*x1[i];
printf("\nw1=w1+d*t*x1=%d",w1);
w2=w2+d*t[i]*x2[i];
printf("\nw2=w2+d*t*x2=%d",w2);
printf("\nNew Matrix of weights is { %d %d %d }",w0,w1,w2);
i++;
yin= w0+((x1[i]*w1)+(x2[i]*w2));
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Name of The Gate=>EX-OR
Enter The Truth Table For the 'EX-OR' Gate
x1    x2    t
0     0     0
0     1     1
1     0     1
1     1     0
yin=0
t=-1
w0=w0+d*t*x0=-1
w1=w1+d*t*x1=1
w2=w2+d*t*x2=1
New Matrix of weights is <-1 1 1>
yin=-1 t=1
w0=w0+d*t*x0=0
w1=w1+d*t*x1=0
w2=w2+d*t*x2=2
New Matrix of weights is <0 0 2>
yin=-1 t=1
w0=w0+d*t*x0=1
w1=w1+d*t*x1=1
w2=w2+d*t*x2=1
New Matrix of weights is <1 1 1>
yin=1 t=-1
w0=w0+d*t*x0=0
w1=w1+d*t*x1=0
w2=w2+d*t*x2=0
New Matrix of weights is <0 0 0>
yin=0 t=-1
Perceptron can't be trained for 'EX-OR' gate
-----
Process exited with return value 0
Press any key to continue . . . _
```

```

yin=yin>0?1:(yin==0?0:-1);
printf("\nyin=%d",yin);
}
printf(" t=%d",t[i]);
if(yin!=t[i])
printf("\nPerceptron can't be trained for '%s' gate" ,b);
else{
printf("\nx1\tx2\tNET");
for(i=0;i<=3;i++)
{
net= (x0*w0)+(x1[i]*w1)+(x2[i]*w2);
net=net<0?0:1;
if(x1[i]==-1)
x1[i]=0;
if(x2[i]==-1)
x2[i]=0;
printf("\n%d\t%d\t%d",x1[i],x2[i],net);
}
printf("\nPerceptron is Trained sucessfully for '%s' Gate\n",b);
}
getch();
}

```



## PRACTICAL NO – 4

**Aim: To design and train a perceptron for NOT gate.**

Program:

```
#include<stdio.h>
void main()
{
int x0=1,d=1,w0=0,w1=0,w2=0,i,j,yin,net;
int x1[4];//={1,1,-1,-1};
//int x2[4];//={1,-1,1,-1};
int t[4];//={1,-1,-1,-1};
int x,y=4;
char b[5];
i=0;
clrscr();
printf("Enter the Name of The Gate=>");
gets(b);
printf("Enter The Truth Table For the '%s' Gate\n",b);
printf("x1\tt\n");
for(j=0;j<2;j++)
{
scanf("%d",&x1[j]);
x=9;
gotoxy(x,y);
//scanf("%d",&x2[j]);
// gotoxy(x+8,y);
scanf("%d",&t[j]);
x=0;y++;
}
for(j=0;j<2;j++)
{ if(x1[j]==0) x1[j]=-1;
// if(x2[j]==0) x2[j]=-1;
if(t[j]==0) t[j]=-1;
}
yin= w0+(x1[0]*w1);//+(x2[0]*w2));
yin=yin>0?1:(yin==0?0:-1);
printf("yin=%d\n",yin);
while((yin!=t[i])&&(i<2))
{
printf(" t=%d",t[i]);
w0=w0+d*t[i]*x0;
printf("\nw0=w0+d*t*x0=%d",w0);
w1=w1+d*t[i]*x1[i];
printf("\nw1=w1+d*t*x1=%d",w1);
//w2=w2+d*t[i]*x2[i];
// printf("\nw2=w2+d*t*x2=%d",w2);
printf("\nNew Matrix of weights is {%d %d}",w0,w1);
i++;
yin= w0+(x1[i]*w1);//+(x2[i]*w2));
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Name of The Gate=>NOT
Enter The Truth Table For the 'NOT' Gate
x1    t
0     1
1     0
yin=0
  t=1
w0=w0+d*t*x0=1
w1=w1+d*t*x1=-1
New Matrix of weights is <1 -1>
  yin=0 t=-1
w0=w0+d*t*x0=0
w1=w1+d*t*x1=-2
New Matrix of weights is <0 -2>
  yin=0 t=2006006826
-----
Perceptron can't be trained for 'NOT' gate
-----
Process exited with return value 0
Press any key to continue . . .
```

```

yin=yin>0?1:(yin==0?0:-1);
printf("\n yin=%d",yin);
}
printf(" t=%d",t[i]);
if(yin!=t[i])
printf("\nPerceptron can't be trained for '%s' gate" ,b);
else{
printf("\nx1\tx2\tNET");
for(i=0;i<=3;i++)
{
net= (x0*w0)+(x1[i]*w1);//(x2[i]*w2);
net=net<0?0:1;
if(x1[i]==-1)
x1[i]=0;
//if(x2[i]==-1)
// x2[i]=0;
printf("\n%d\t%d",x1[i],net);
}
printf("\nPerceptron is Trained sucessfully for '%s' Gate\n",b);
}
getch();
}

```





## PRACTICAL NO- 5

**Aim: To design and train a perceptron for identifying ODD and EVEN number.**

Program:

```
#include<stdio.h>
void main()
{
int x0=1,d=1,w0=0,w1=0,w2=0,w3=0,w4=0,i,j,yin,net,net1;
int x1[10]={-1,-1,-1,-1,-1,-1,-1,-1,1,1};
int x2[10]={-1,-1,-1,-1,1,1,1,1};
int x3[10]={-1,-1,1,1,-1,-1,1,1,-1,-1};
int x4[]={-1,1,-1,1,-1,1,-1,1,-1,1};
int t[]={-1,1,-1,1,-1,1,-1,1,-1,1};
int x,y=4;
int bin[5],k;
//char b[5];
i=0;
clrscr();
printf("Enter the Value=> ");
scanf("%d",&x);
yin= w0+((x1[0]*w1)+(x2[0]*w2)+(x3[0]*w3)+(x4[0]*w4));
yin=yin>0?1:(yin==0?0:-1);
printf("yin=%d\n",yin);
while((yin!=t[i])&&(i<10))
{
printf(" t=%d",t[i]);
w0=w0+d*t[i]*x0;
printf("\nw0=w0+d*t*x0=%d",w0);
w1=w1+d*t[i]*x1[i];
printf("\nw1=w1+d*t*x1=%d",w1);
w2=w2+d*t[i]*x2[i];
printf("\nw2=w2+d*t*x2=%d",w2);
w3=w3+d*t[i]*x3[i];
printf("\nw3=w3+d*t*x3=%d",w3);
w4=w4+d*t[i]*x4[i];
printf("\nw4=w4+d*t*x4=%d",w4);
printf("\nNew Matrix of weights is { %d %d %d %d %d }",w0,w1,w2,w3,w4);
i++;
yin= w0+((x1[i]*w1)+(x2[i]*w2)+(x3[i]*w3)+(x4[i]*w4));
yin=yin>0?1:(yin==0?0:-1);
printf("\nyin=%d",yin);
}
printf(" t=%d",t[i]);
if(yin!=t[i])
printf("\nPerceptron can't be trained for classifying ODD and EVEN numbers");
else{

y=x%10;
k=0;
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Value=> 23789
yin=0
t=-1
w0=w0+d*t*x0=-1
w1=w1+d*t*x1=1
w2=w2+d*t*x2=1
w3=w3+d*t*x3=1
w4=w4+d*t*x4=1
New Matrix of weights is <-1 1 1 1 1>
yin=-1 t=1
w0=w0+d*t*x0=0
w1=w1+d*t*x1=0
w2=w2+d*t*x2=0
w3=w3+d*t*x3=0
w4=w4+d*t*x4=2
New Matrix of weights is <0 0 0 0 2>
yin=-1 t=-11001
The Entered Number is ODD
-----
Process exited with return value 0
Press any key to continue . . .
```

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the Value=> 2346
yin=0
t=-1
w0=w0+d*t*x0=-1
w1=w1+d*t*x1=1
w2=w2+d*t*x2=1
w3=w3+d*t*x3=1
w4=w4+d*t*x4=1
New Matrix of weights is <-1 1 1 1 1>
yin=-1 t=1
w0=w0+d*t*x0=0
w1=w1+d*t*x1=0
w2=w2+d*t*x2=0
w3=w3+d*t*x3=0
w4=w4+d*t*x4=2
New Matrix of weights is <0 0 0 0 2>
yin=-1 t=-1011
The Entered Number is Even
-----
Process exited with return value 0
Press any key to continue . . .
```

```

if(y==0)
{
bin[0]=0;
bin[1]=0;
bin[2]=0;
bin[3]=0;
}
else
{
while(y!=0)
{
bin[k]=y%2;
y=y/2;
printf("%d",bin[k]);
k++;
}
}
net= ((x0*w0)+(bin[3]*w1)+(bin[2]*w2)+(bin[1]*w3)+(bin[0]*w4));
// printf("\n%d",net);
net1=net>0?1:0;
// printf("\n%d",net1);
if (net1 == 0)
printf(" \nThe Entered Number is Even");
if (net1 == 1)
printf("\n The Entered Number is ODD");
}

getch();
}

```



## PRACTICAL NO- 6

**Aim: To create a Bi-directional Associative Memory (BAM) for ID and telephone number.**

Program:

```
void main()
{
int o[25][25],newid,orgno,tw;
int a1[5][25],a2[15][25],a3[15][25];
int b1[15][25],b2[15][25],b3[55][25],a1t[25][5],a2t[25][5];
int a3t[25][5],w[25][25],wt[25][25],i,j,k,id[5],tele[5];
int bin_id[5][15];
int bin_tele[15][25],t;
int n[10],t1[10],temp1[8][8],temp2[8][8],temp3[8][8],temp[8][8],bin_temp[8][8];
clrscr();
for(i=0;i<3;i++)
{printf("ENter the idfor telephone no=> ");
scanf("%d",&id[i]);
printf("Enter the telephone no=> ");
scanf("%d",&tele[i]);
}
for(i=0;i<3;i++)
{
t=id[i];
k=-1;
while(t>0)
{
k++;
bin_id[i][k]=t%2;
t=t/2;
}
n[i]=k;
printf("\nTHE BINARY IS ");
for(j=k;j>=0;j--)
{printf("%d",bin_id[i][j]);}
}
for(i=0;i<3;i++)
{
t=tele[i];
k=-1;
while(t>0)
{
k++;
bin_tele[i][k]=t%2;
t=t/2;
}
t1[i]=k;
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the id for telephone no=> 12
Enter the telephone no=> 23456
Enter the id for telephone no=> 13
Enter the telephone no=> 23457
Enter the id for telephone no=> 14
Enter the telephone no=> 23458

THE BINARY IS 1100
THE BINARY IS 1101
THE BINARY IS 1110
ait*b1=>

Weight matrix is (ait*b1+a2t*b2+a3t*b3)=>
  3  -3  3  3  -3  3  3  3  -3  3  -3  -3  -3  -1  -1
  3  -3  3  3  -3  3  3  3  -3  3  -3  -3  -3  -1  -1
 -1  1  -1  -1  1  -1  -1  -1  -1  1  -1  1  1  3  -1
 -1  1  -1  -1  1  -1  -1  -1  -1  1  -1  1  1  -1  3

Enter the id for telephone no 12
6 -6 6 6 -6 6 6 6 -6 6 -6 -6 -6 -2 -2
The telephone n0 is 101101110100000
The Telephone no is=> 23456

-----
Process exited with return value 0
Press any key to continue . . . _
```

```

for(i=n[0],k=0;i>=0;i--,k++)
{
a1[0][k]=bin_id[0][i];
if(a1[0][k]==0){a1[0][k]=-1;}
}
for(i=n[1],k=0;i>=0;i--,k++)
{
a2[0][k]=bin_id[1][i];
if(a2[0][k]==0){a2[0][k]=-1;}
}

for(i=n[2],k=0;i>=0;i--,k++)
{
a3[0][k]=bin_id[2][i];
if(a3[0][k]==0){a3[0][k]=-1;}
}

for(i=t1[0],k=0;i>=0;i--,k++)
{
b1[0][k]=bin_tele[0][i];
if(b1[0][k]==0){ b1[0][k]=-1;}
//printf(" b1[0][%d]=%d",k,b1[0][k]);
}
for(i=t1[1],k=0;i>=0;i--,k++)
{
b2[0][k]=bin_tele[1][i];
if( b2[0][k]==0){b2[0][k]=-1;}
// printf(" b2[0][%d]=%d",k,b2[0][k]);
}
for(i=t1[2],k=0;i>=0;i--,k++)
{
b3[0][k]=bin_tele[2][i];
if(b3[0][k]==0){b3[0][k]=-1;}
// printf(" b3[0][%d]=%d",k,b3[0][k]);
}

for(i=0;i<=n[0];i++)
{
a1t[i][0]=a1[0][i]; //transpose of a1
//printf("a1t[%d][0]=%d",i,a1t[i][0]);
}

for(i=0;i<=n[1];i++)
{
a2t[i][0]=a2[0][i]; //transpose of a1
// printf("a2t[%d][0]=%d",i,a2t[i][0]);
}
for(i=0;i<=n[2];i++)
{
a3t[i][0]=a3[0][i]; //trsanspose of a1
// printf("a3t[%d][0]=%d",i,a3t[i][0]);
}

```





```

//calculating weight matrix w=a1t*b1+a2t*b2+a3t*b3
printf("a1t*b1=>\n");
for(i=0;i<=n[0];i++)
{ printf("\n");
for(j=0;j<=t1[0];j++)
{ temp1[i][j]=0;
temp1[i][j]+=a1t[i][0]*b1[0][j];
temp2[i][j]=0;
temp2[i][j]+=a2t[i][0]*b2[0][j];
temp3[i][j]=0;
temp3[i][j]+=a3t[i][0]*b3[0][j];
w[i][j]=0;
w[i][j]=temp1[i][j]+temp2[i][j]+temp3[i][j];
}
}
printf(" Weight matrix is (a1t*b1+a2t*b2+a3t*b3)=> \n");
for(i=0;i<=n[0];i++ )
{
printf("\n");
for(j=0;j<=t1[0];j++)
{
printf(" %d",w[i][j]);
}}

printf("\nENter the id for telephone no");
scanf("%d",&newid);
i=-1;
while(newid>0)
{
i++;
temp[0][i]=newid%2;
newid/=2;
}

for(k=i,j=0;k>=0;k--,j++)
{
bin_temp[0][j]=temp[0][k];
}
for(i=0;i<=t1[0];i++)
{
for(j=0;j<=n[0];j++)
{ o[i][j]=0;
o[0][i]+=bin_temp[0][j]*w[j][i];
}
}
for(i=0;i<=t1[0];i++)
{
printf(" %d",o[0][i]);
}
}

```



```

for(i=0;i<=t1[0];i++)
{
    o[0][i]=o[0][i]<=0?0:1;
}

printf("The telephone nO is ");
for(i=0;i<=t1[0];i++)
{
    printf("%d",o[0][i]);
}

orgno=0;
for(i=0;i<=t1[0];i++)
{ tw=1;
  for(k=0;k<t1[0]-(i);k++)
  {
    tw*=2;
    // printf(" tw %d",tw);
  }

  orgno+=o[0][i]*tw;
}
printf("The Telephone no is=> %d",orgno);
getch();
}

```



## PROGRAM NO -7

**Aim: To design and train the Hopfield net to map the input vector with the stored vector and correct them.**

Program:

```
void main()
{
int a[5][5],b[5][5],at[5][5],w[5][5],n,i,j,k,x[5][5],y[5][5],yin;
clrscr();

printf("Enter the no of values in the Stored vector=>");
scanf("%d",&n);
printf("ENter the valued in the Stored vector=>");
for(i=0;i<n;i++)
{scanf("%d",&a[0][i]);}
for(i=0;i<n;i++)
{
if(a[0][i]==0)
{a[0][i]=-1;}
for(i=0;i<n;i++)
{at[i][0]=a[0][i];}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{ w[i][j]=0;
w[i][j]+=at[i][0]*a[0][j];
} }
for(i=0;i<n;i++)
w[i][i]=0; //initilizing the w[i][i]=0
printf("THE weight Matric is ");
for(i=0;i<n;i++)
{printf("\n");
for(j=0;j<n;j++)
{printf("\t%d",w[i][j]);}}

printf("\nEnter the New vector=>");
for(i=0;i<n;i++)
{scanf("%d",&x[0][i]);
y[0][i]=x[0][i];
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
yin+=(y[0][j]*w[j][i]);
yin+=x[0][i];
yin=yin<=0?0:1;
y[0][i]=yin;
}
}
```

## Output:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Enter the no of values int he Stored vector=>5
ENter the valued in the Stored vector=>1
0
1
1
1
0
The weight Matric is
  0      -1      1      1      -1
 -1      0     -1     -1      1
  1     -1      0      1     -1
  1     -1      1      0     -1
 -1      1     -1     -1      0
Enter the New vector=>0
0
1
1
1
0
The value of the new vector should br=> 1 0 1 1 0
-----
Process exited with return value 0
Press any key to continue . . . _
```

```
printf("The value of the new vector should be=> ");  
for(i=0;i<n;i++)  
{  
printf(" %d",y[0][i]);  
}  
getch();  
}
```