**Program 1:- Explain the following five networking commands :**

1. **Command Name :** Ping

**Purpose:** Test the network connection with a remote IP Address.

**Output :-**

C:\Users\Shubham>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]

[-r count] [-s count] [[-j host-list] | [-k host-list]]

[-w timeout] [-R] [-S srcaddr] [-4] [-6] target\_name

Options:

-t Ping the specified host until stopped.

To see statistics and continue - type Control-Break;

To stop - type Control-C.

-a Resolve addresses to hostnames.

-n count Number of echo requests to send.

-l size Send buffer size.

-f Set Don't Fragment flag in packet (IPv4-only).

-i TTL Time To Live.

-v TOS Type Of Service (IPv4-only. This setting has been deprecated

and has no effect on the type of service field in the IP Header).

-r count Record route for count hops (IPv4-only).

-s count Timestamp for count hops (IPv4-only).

-j host-list Loose source route along host-list (IPv4-only).

-k host-list Strict source route along host-list (IPv4-only).

-w timeout Timeout in milliseconds to wait for each reply.

-R Use routing header to test reverse route also (IPv6-only).

-S srcaddr Source address to use.

-4 Force using IPv4.

-6 Force using IPv6.

**Command Name :- ping 192.168.3.43**

**Output :-**

C:\Users\user16>ping 192.168.3.43

Pinging 192.168.3.43 with 32 bytes of data:

Reply from 192.168.3.43: bytes=32 time<1ms TTL=128

Reply from 192.168.3.43: bytes=32 time<1ms TTL=128

Reply from 192.168.3.43: bytes=32 time<1ms TTL=128

Reply from 192.168.3.43: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.3.43:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

1. **Command Name :** IpConfig

**Purpose:** Display or refresh the TCP/IP configuration.

**Output :-**

C:\Users\Shubham>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :

Link-local IPv6 Address . . . . . : fe80::1586:9096:cfb8:336d%11

IPv4 Address. . . . . . . . . . . : 192.168.1.169

Subnet Mask . . . . . . . . . . . : 255.255.255.0

Default Gateway . . . . . . . . . : 192.168.1.230

Tunnel adapter isatap.{8B04009C-0678-452F-8B11-B5B8EF3E97A9}:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Examples:

> ipconfig ... Show information

> ipconfig /all ... Show detailed information

> ipconfig /renew ... renew all adapters

> ipconfig /renew EL\* ... renew any connection that has its

name starting with EL

> ipconfig /release \*Con\* ... release all matching connections,

eg. "Local Area Connection 1" or

"Local Area Connection 2"

> ipconfig /allcompartments ... Show information about all

compartments

> ipconfig /allcompartments /all ... Show detailed information about all

Compartments

1. **Command Name :** Netstat

**Purpose:**  Display the status of the TCP/IP stack on the local machine.

**Output :-**

C:\Users\msc 6>netstat

Active Connections

Proto Local Address Foreign Address State

TCP 192.168.1.169:49284 pc-b:http CLOSE\_WAIT

TCP 192.168.1.169:49354 14:8080 CLOSE\_WAIT

TCP 192.168.1.169:49483 13.78.132.29:http ESTABLISHED

\

1. **Command Name :** Telnet

**Purpose:**  The telnet command to access to a remote host in terminal mode.

**Output :-**

Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> CTRL+j

Invalid Command. type ?/help for help

Microsoft Telnet> ?/HELP

Commands may be abbreviated. Supported commands are:

c - close close current connection

d - display display operating parameters

o - open hostname [port] connect to hostname (default port 23).

q - quit exit telnet

set - set set options (type 'set ?' for a list)

sen - send send strings to server

st - status print status information

u - unset unset options (type 'unset ?' for a list)

?/h - help print help information

Microsoft Telnet> O

( to ) 192.168.3.43

Connecting To 192.168.3.43...Could not open connection to the host, on port 23: Connect failed

1. **Command Name :** Tracert

**Purpose:**  Display all intermediate IP Addresses through which a packet passes through, between the local machine and the specified IP Address.

**Output :-**

**C:\Users\Shubham>tracert**

Usage: tracert [-d] [-h maximum\_hops] [-j host-list] [-w timeout]

[-R] [-S srcaddr] [-4] [-6] target\_name

Options:

-d Do not resolve addresses to hostnames.

-h maximum\_hops Maximum number of hops to search for target.

-j host-list Loose source route along host-list (IPv4-only).

-w timeout Wait timeout milliseconds for each reply.

-R Trace round-trip path (IPv6-only).

-S srcaddr Source address to use (IPv6-only).

-4 Force using IPv4.

-6 Force using IPv6.

**C:\Users\Shubham>tracert 192.168.1.169**

Tracing route to Shubham [192.168.1.169] over a maximum of 30 hops:

1 <1 ms <1 ms <1 ms Shubham [192.168.1.169]

Trace complete.

C:\Users\ Shubham >tracert 169.254.234.30

Tracing route to 169.254.234.30 over a maximum of 30 hops

1 Shubham [192.168.1.169] reports: Destination host unreachable.

Trace complete.

C:\Users\ Shubham >tracert google.com

Tracing route to google.com [172.217.26.206]

over a maximum of 30 hops:

1 <1 ms <1 ms <1 ms 192.168.1.230

2 \* \* \* Request timed out.

3 \* \* \* Request timed out.

4 \* \* \* Request timed out.

**Program:- 2.**

**Write a program in C++ to implement Dijkstra’s algorithm to compute shortest path through a group.**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#define MAXNODES 20

#define INFINITY 999

class Shortpath

{

int weight[MAXNODES][MAXNODES],i,j,distance[MAXNODES],visit[MAXNODES];

int precede[MAXNODES],final;

int path[140];

int smalldist,newdist,k,s,d,current,n,distcurr;

public:

void display();

void Dijkstra();

};

void Shortpath::Dijkstra()

{

cout<<"\nEnter the number of nodes:";

cin>>n;

cout<<"\nEnter the cost of matrix:\n\n";

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

{

cin>>weight[i][j];

}

}

cout<<"\nEnter the source node(1 to"<<n<<"):";

cin>>s;

cout<<"\nEnter the destination node(1 to"<<n<<"):";

cin>>d;

for(i=1;i<=n;i++)

{

distance[i]=INFINITY;

precede[i]=INFINITY;

}

distance[s]=0;

current=s;

visit[current]=1;

while(current!=d)

{

distcurr=distance[current];

smalldist=INFINITY;

for(i=1;i<=n;i++)

if(visit[i]==0)

{

newdist=distcurr+weight[current][i];

if(newdist<distance[i])

{

distance[i]=newdist;

precede[i]=current;

}

if(distance[i]<smalldist)

{

smalldist=distance[i];

k=i;

}

}

current=k;

visit[current]=1;

}

}

void Shortpath::display()

{

final=0;

i=d;

path[final]=d;

final++;

while(precede[i]!=s)

{

j=precede[i];

i=j;

path[final]=i;

final++;

}

path[final]=i;

final++;

path[final]=s;

cout<<"\nThe shorter path followed is:\n\n";

for(i=final;i>0;i--)

{

cout<<"\t"<<path[i]<<"\_>"<<path[i-1];

}

cout<<"\nFor total cost="<<distance[d];

}

void main()

{

clrscr();

Shortpath s;

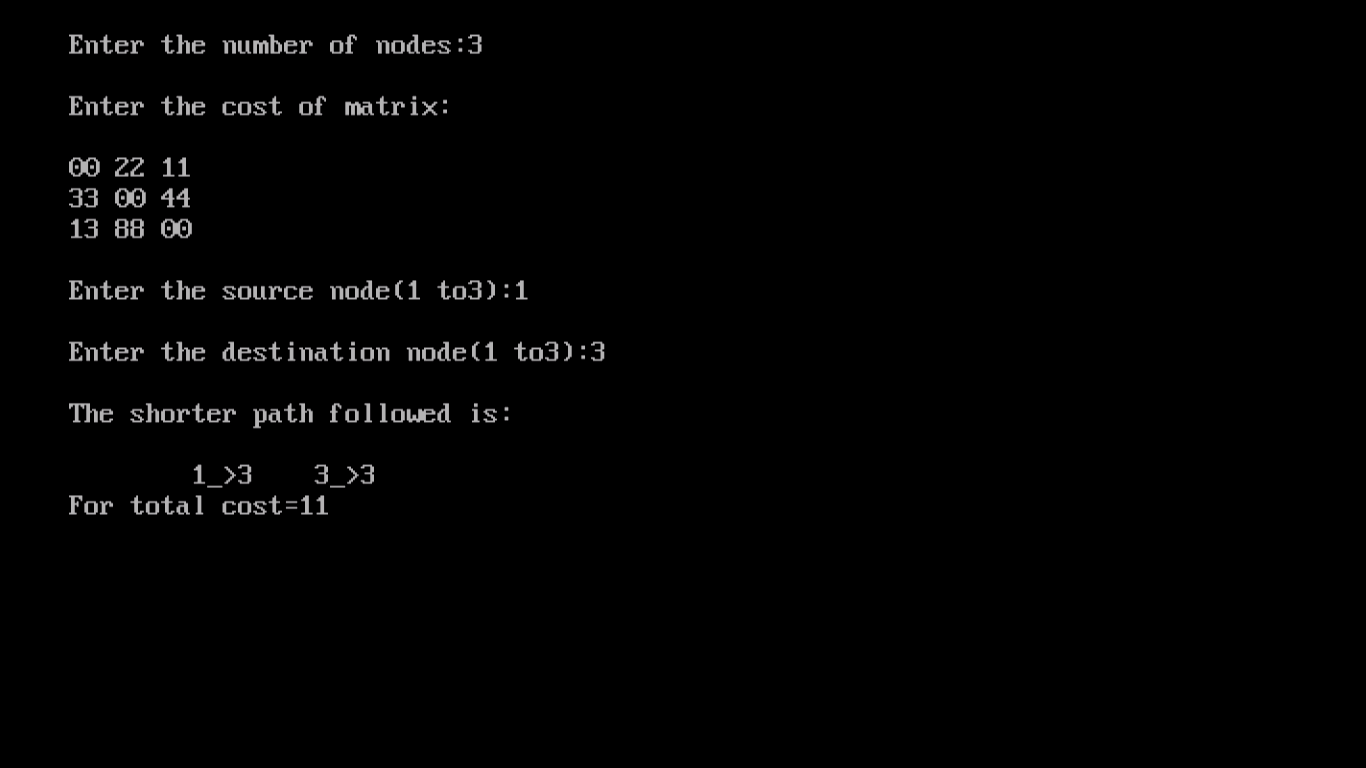
s.Dijkstra();

s.display();

getch();

}

**Output :-**



**Program :- 3.**

**Write a program in C++ to implement of distance vector routing algorithm.**

#include<iostream.h>

#include<conio.h>

char ch[7]={'A','B','C','D','E','F','G'};

class dvr

{

int graph[50][50],via[50][50],final[50][50];

int i,j,k,t;

int nn;

public:

void getdata();

void sharing();

void display();

};

void dvr::getdata()

{

cout<<"\n Enter number of nodes:";

cin>>nn;

for(i=0;i<nn;i++)

{

for(j=0;j<nn;j++)

{

graph[i][j]=-1;

}

}

for(i=0;i<nn;i++)

{

for(j=0;j<nn;j++)

{

if(i==j)

{

graph[i][j]=0;

}

if(graph[i][j]==-1)

{

cout<<"\n Enter distance between"<<ch[i]<<"-"<<ch[j]<<"-";

cin>>t;

graph[i][j]=graph[j][i]=t;

}

}

}

getch();

for(i=0;i<nn;i++)

{

for(j=0;j<nn;j++)

{

via[i][j]=-1;

}

}

cout<<"\n After initialization:";

for(i=0;i<nn;i++)

{

cout<<"\n"<<ch[i]<<"Table";

cout<<"\n Node\t Dist\t Via:";

for(j=0;j<nn;j++)

{

cout<<"\n"<<ch[j]<<"\t"<<graph[i][j]<<"\t"<<via[i][j];

}

}

getch();

}

void dvr::sharing()

{

int sh[20][20][20];

for(i=0;i<nn;i++)

{

for(j=0;j<nn;j++)

{

for(k=0;k<nn;k++)

{

if((graph[i][j]>-1)&&(graph[i][j]>-1))

{

sh[i][j][k]=graph[j][k]+graph[i][j];

}

else

{

sh[i][j][k]=-1;

}

}

}

}

for(i=0;i<nn;i++)

{

cout<<"\n\n for"<<ch[i];

for(j=0;j<nn;j++)

{

cout<<"\n From"<<ch[j];

for(k=0;k<nn;k++)

{

cout<<"\n"<<ch[k]<<"\t"<<sh[i][j][k];

}

}

getch();

}

getch();

for(i=0;i<nn;i++)

{

for(j=0;j<nn;j++)

{

final[i][j]=graph[i][k];

via[i][j]=i;

for(k=0;k<nn;k++)

{

if((final[i][j]>sh[i][j][k])||(final[i][j]++-1))

{

if(sh[i][k][j]>-1)

{

final[i][j]=sh[i][k][j];

via[i][j]=k;

}

}

}

if(final[i][j]==-1)

{

for(k=0;k<nn;k++)

{

if((final[i][k]!=-1)&&(final[k][i]!=-1))

{

if((final[i][j]==-1)||((final[i][j]!=-1)&&(final[i][j]>final[i][k]+final[k][j])))

{

if(final[i][k]>final[k][j]>-1)

{

final[i][j]=final[i][k]+final[k][j];

via[i][j]=k;

}

}

}

}

}

}

}

}

void dvr::display()

{

for(i=0;i<nn;i++)

{

cout<<"\n"<<ch[i]<<"Table";

cout<<"\n Node \t Dist \t via";

for(j=0;j<nn;j++)

{

cout<<"\n"<<ch[j]<<"\t"<<final[i][j]<<"\t";

if(i==via[i][j])

cout<<"-";

else

cout<<ch[via[i][j]];

}

}

getch();

}

void main()

{

clrscr();

dvr d1;

d1.getdata();

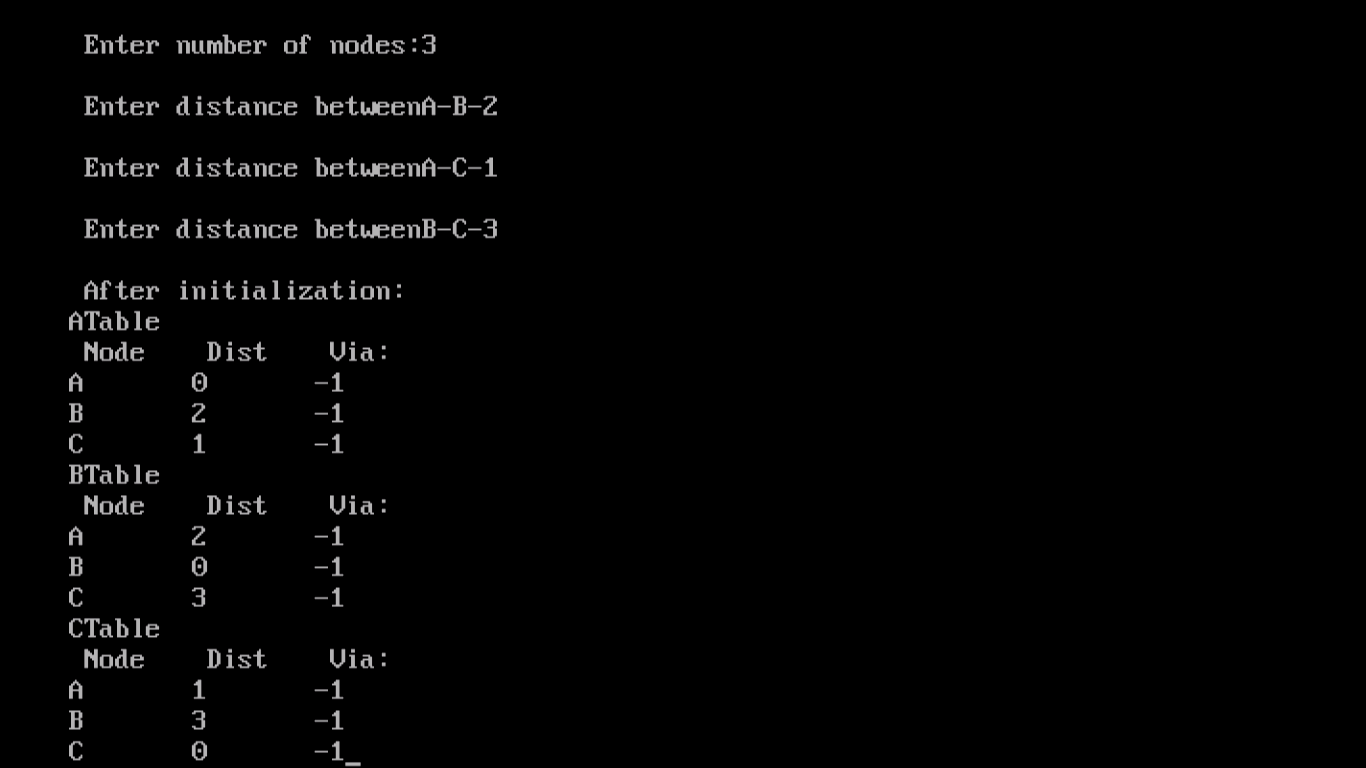
d1.sharing();

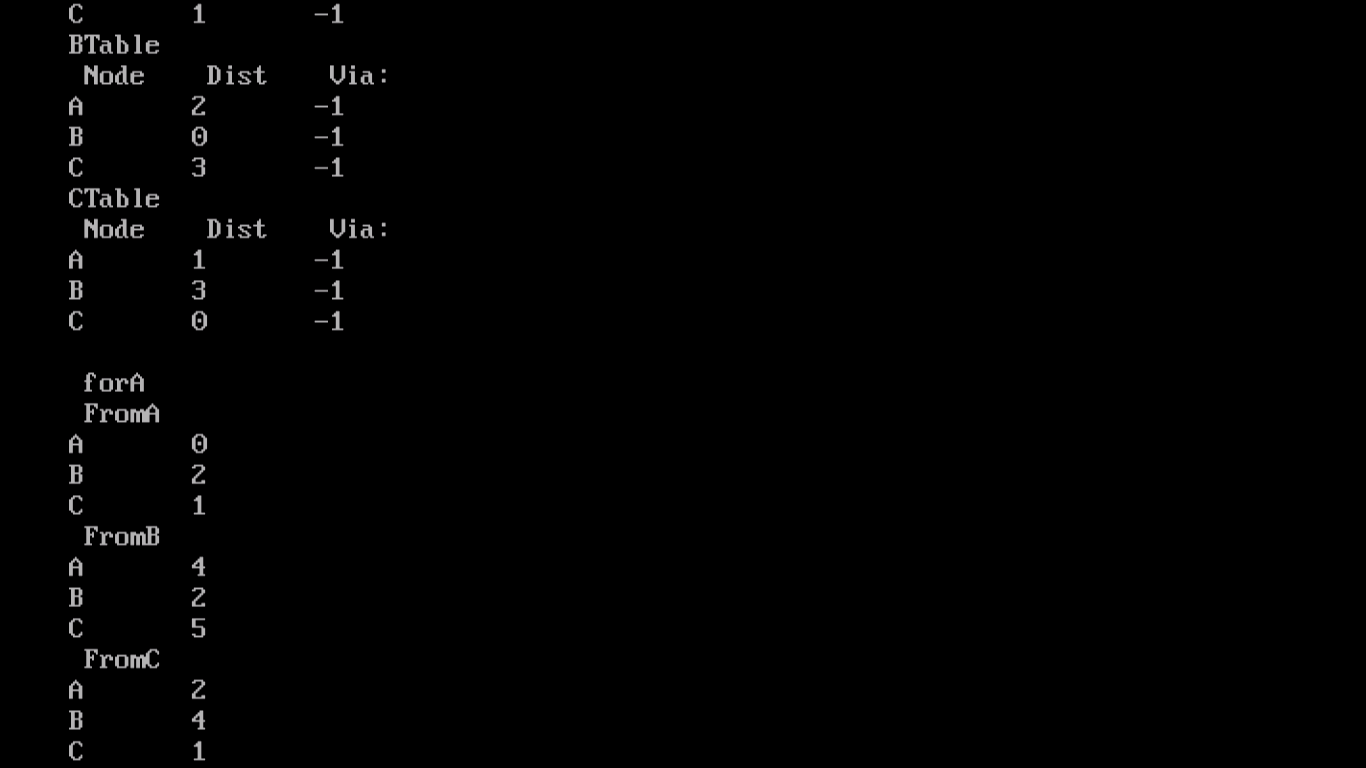
d1.display();

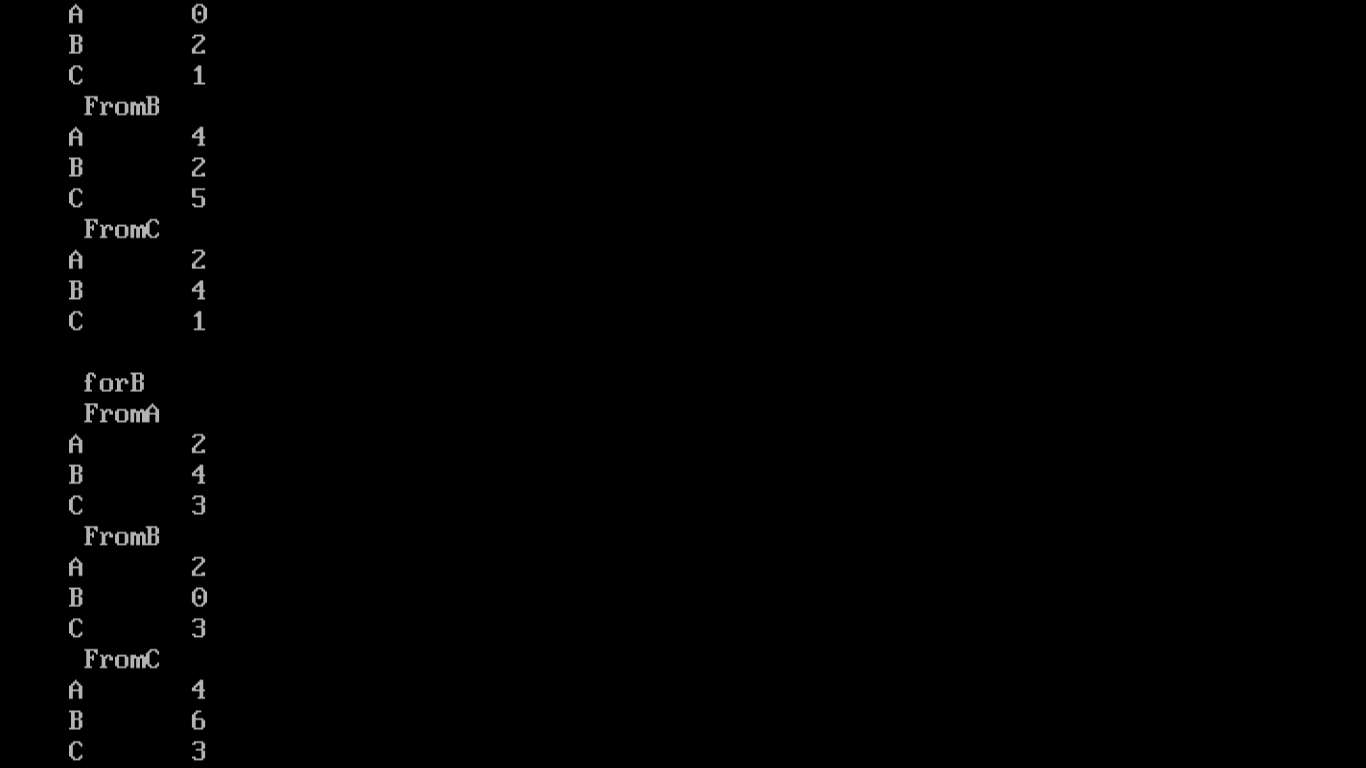
getch();

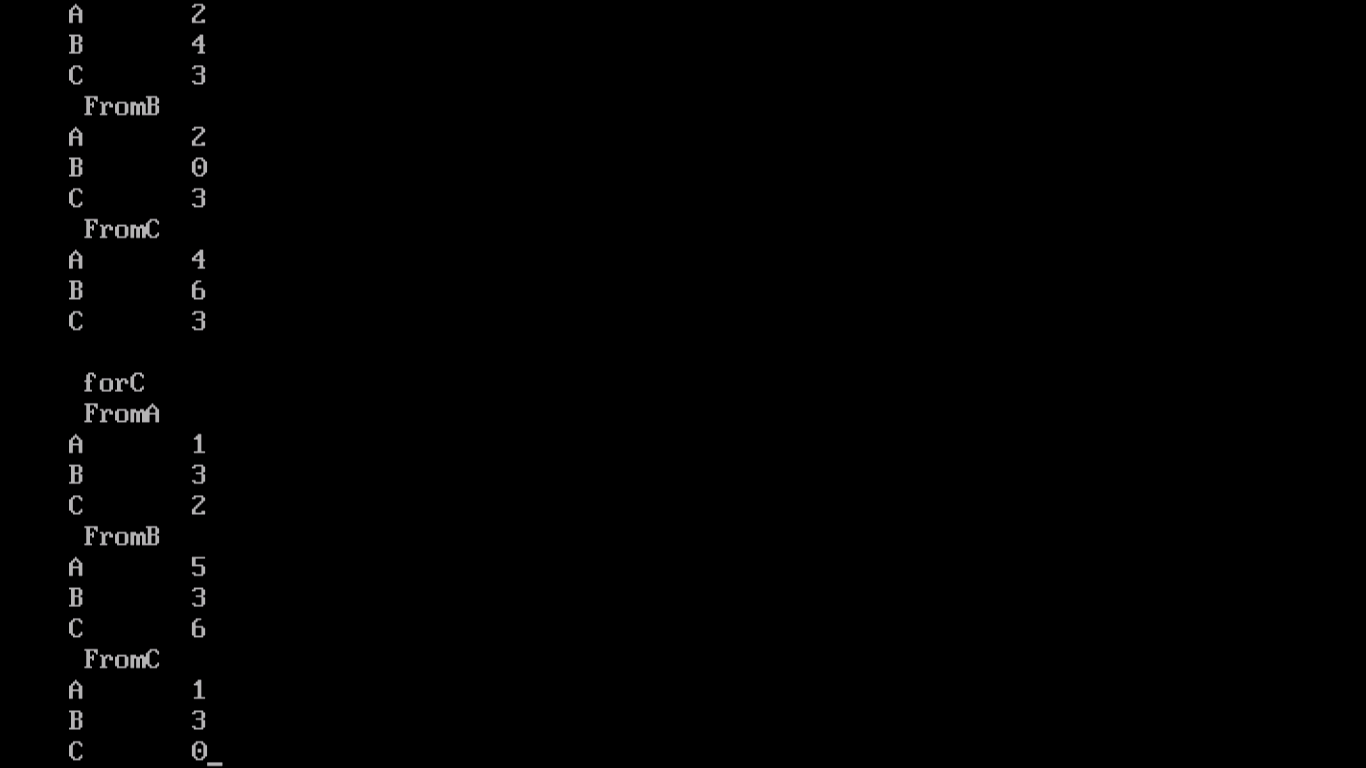
}

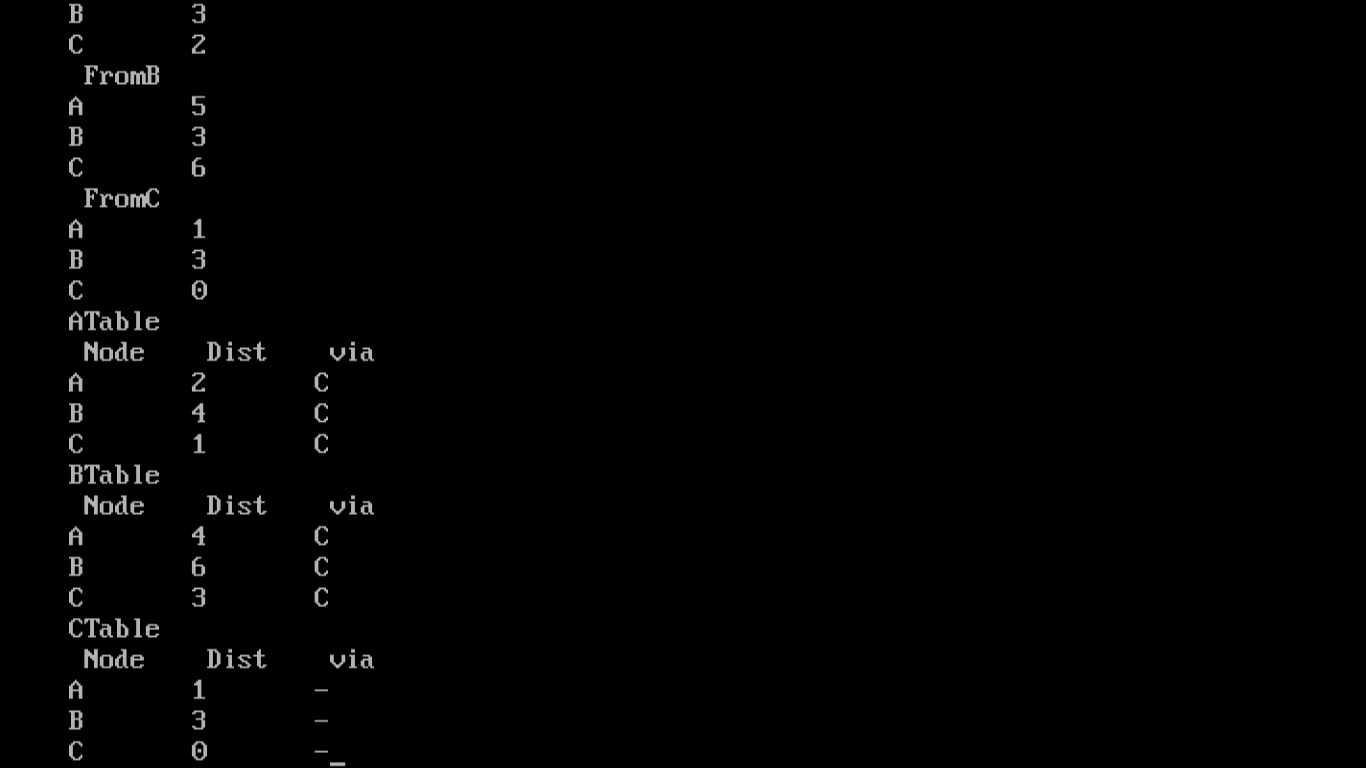
**Output :-**











**Program :- 4.**

**Write a program in C++ to find the cyclic redundancy check for the given Data.**

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

#include<string.h>

#define N strlen(g)

char t[28],cs[28],g[]="100001000000100001";

int a, e, c;

void xor()

{

for(c=1;c<N;c++)

cs[c]=((cs[c]==g[c]?'0':'1'));

}

void crc()

{

for(e=0;e<N;e++)

cs[e]=t[e];

do

{

if(cs[0]=='1')

xor();

for(c=0;c<N-1;c++)

cs[c]=cs[c+1];

cs[c]=t[e++];

}

while(e<=a+N-1);

}

int main()

{

clrscr();

cout<<"\nEnter the data:";

cin>>t;

cout<<"\n-------------------------------";

cout<<"\n Generating Polynomial:"<<g;

a=strlen(t);

for(e=a;e<a+N-1;e++)

t[e]='0';

cout<<"\n--------------------------------";

cout<<"\n Modified data is:"<<t;

cout<<"\n Checksum is:"<<cs;

for(e=a;e<a+N-1;e++)

t[e]=cs[e-a];

cout<<"\n--------------------------------";

cout<<"\n Final codeword is:"<<t;

cout<<"\n--------------------------------";

cout<<"\n Test error detection 0(yes) 1(no)?:";

cin>>e;

if(e==0)

{

do

{

cout<<"\n Enter the position where error is to be inserted:";

cin>>e;

}

while(e==0||e>(a+N-1));

t[e-1]=(t[e-1]=='0')?'1':'0';

cout<<"\n--------------------------------";

cout<<"\n Erroneous data:"<<t;

}

crc();

for(e=0;(e<N-1)&&(cs[e]!='1');e++);

if(e<N-1)

cout<<"\n Error detected\n\n";

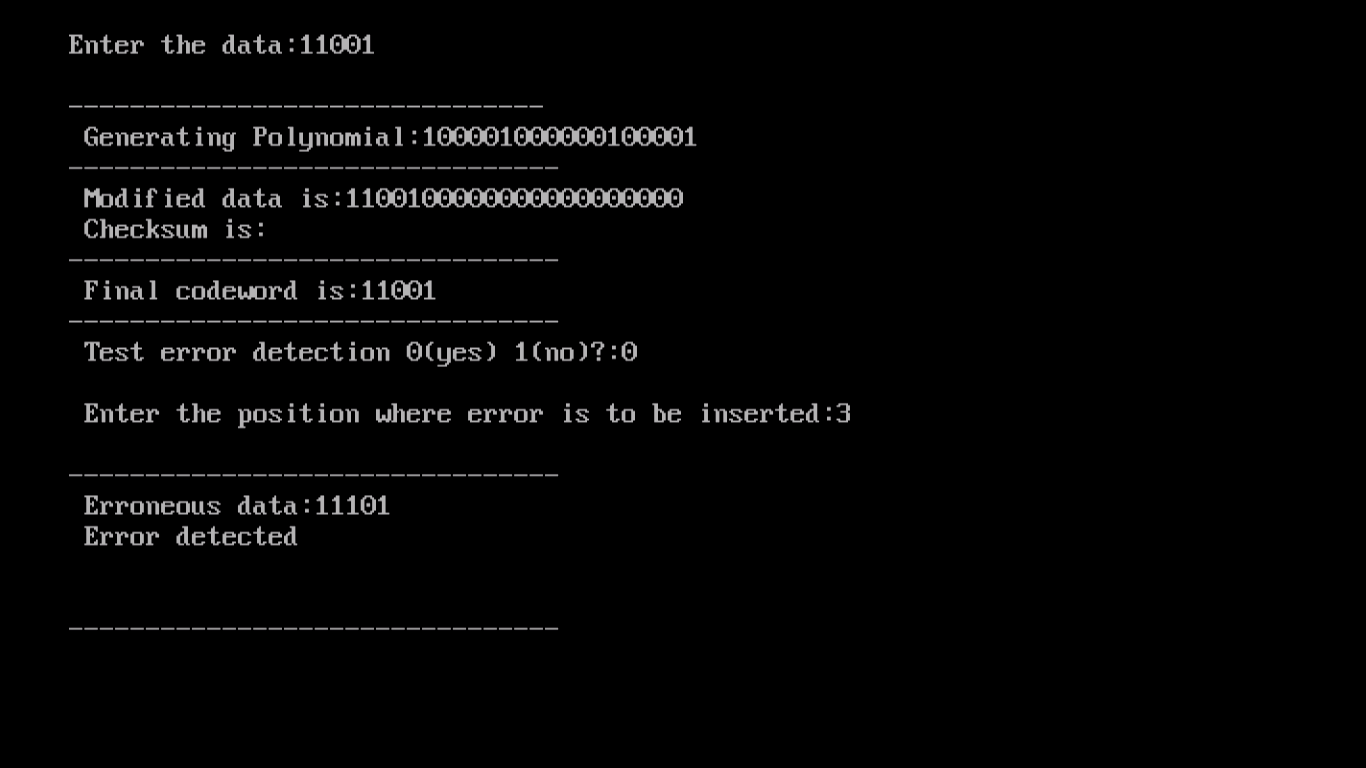
cout<<"\n--------------------------------";

getch();

return 0;

}

**Output :-**



**Program 5**

**Write a program in C++ to perform simulation on sliding window protocol.**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

#include<math.h>

#define TOT\_FRAMES 500

#define FRAMES\_SEND 10

class sel\_repeat

{

private:

int fr\_send\_at\_instance;

int arr[TOT\_FRAMES];

int send[FRAMES\_SEND];

int rcvd[FRAMES\_SEND];

char rcvd\_ack[FRAMES\_SEND];

int sw;

int rw;

public:

void input();

void sender(int);

void receiver(int);

};

void sel\_repeat::input()

{

int n;

int m;

int i;

cout<<"\n Enter the no. of bits for the sequence no. :";

cin>>n;

m=pow(2,n);

int t=0;

fr\_send\_at\_instance=(m/2);

for(i=0; i<TOT\_FRAMES; i++)

{

arr[i]=t;

t=(t+1)%m;

}

for(i=0; i<fr\_send\_at\_instance; i++)

{

send[i]=arr[i];

rcvd[i]=arr[i];

rcvd\_ack[i]='n';

}

rw=sw=fr\_send\_at\_instance;

sender(m);

}

void sel\_repeat::sender(int m)

{

for(int i=0; i<fr\_send\_at\_instance; i++)

{

if(rcvd\_ack[i]=='n')

cout<<"SENDER:Frame"<<send[i]<<"is sent \n";

}

receiver(m);

}

void sel\_repeat::receiver(int m)

{

time\_t t;

int f, j, fl, al, k;

char ch;

srand((unsigned)time(&t));

for(int i=0; i<fr\_send\_at\_instance; i++)

{

if(rcvd\_ack[i]=='n')

{

f=rand()%10;

if(f!=5)

{

for(int j=0; j<fr\_send\_at\_instance; j++)

if(rcvd[j]==send[i])

{

cout<<"receiver:Frame"<<rcvd[j]<<"received correctly \n";

rcvd[j]=arr[rw];

rw=(rw+1)%m;

break;

}

if(j==fr\_send\_at\_instance)

{

cout<<"\n Receiver:Duplicate frame"<<send[i]<<"discarded \n";

}

int al=rand()%5;

if(al==3)

{

cout<<"(acknowledgement"<<send[i]<<"lost) \n";

cout<<"(sender timeouts -->Resend the frame)\n";

rcvd\_ack[i]='n';

}

else

{

cout<<"(acknowledgement"<<send[i]<<"received)\n";

rcvd\_ack[i]='p';

}

}

else

{

int ld=rand()%2;

if(ld==0)

{

cout<<"\n RECEIVER:Frame"<<send[i]<<"is damaged \n";

cout<<"\n RECEIVER:Negative acknowledgement"<<send[i]<<"sent \n";

}

else

{

cout<<"Receiver frame"<<send[i]<<"is lost \n";

cout<<"(SENDER TIMEOUTS-->>RESEND THE FRAME)\n";

}

rcvd\_ack[i]='n';

}}}

for(j=0; j<fr\_send\_at\_instance; j++)

{

if(rcvd\_ack[i]=='n')

break;

}

i=0;

for(k=j; k<fr\_send\_at\_instance; k++)

{

send[i]=send[k];

if(rcvd\_ack[k]=='n')

rcvd\_ack[i]='n';

else

rcvd\_ack[i]='p';

i++;

}

if(i!=fr\_send\_at\_instance)

{

for(k=i; k<fr\_send\_at\_instance; k++)

{

send[k]=arr[sw];

sw=(sw+1)%m;

rcvd\_ack[k]='n';

}

}

cout<<"Want to continue";

cin>>ch;

cout<<"\n";

if(ch=='y')

sender(m);

else

exit(0);

}

void main()

{

clrscr();

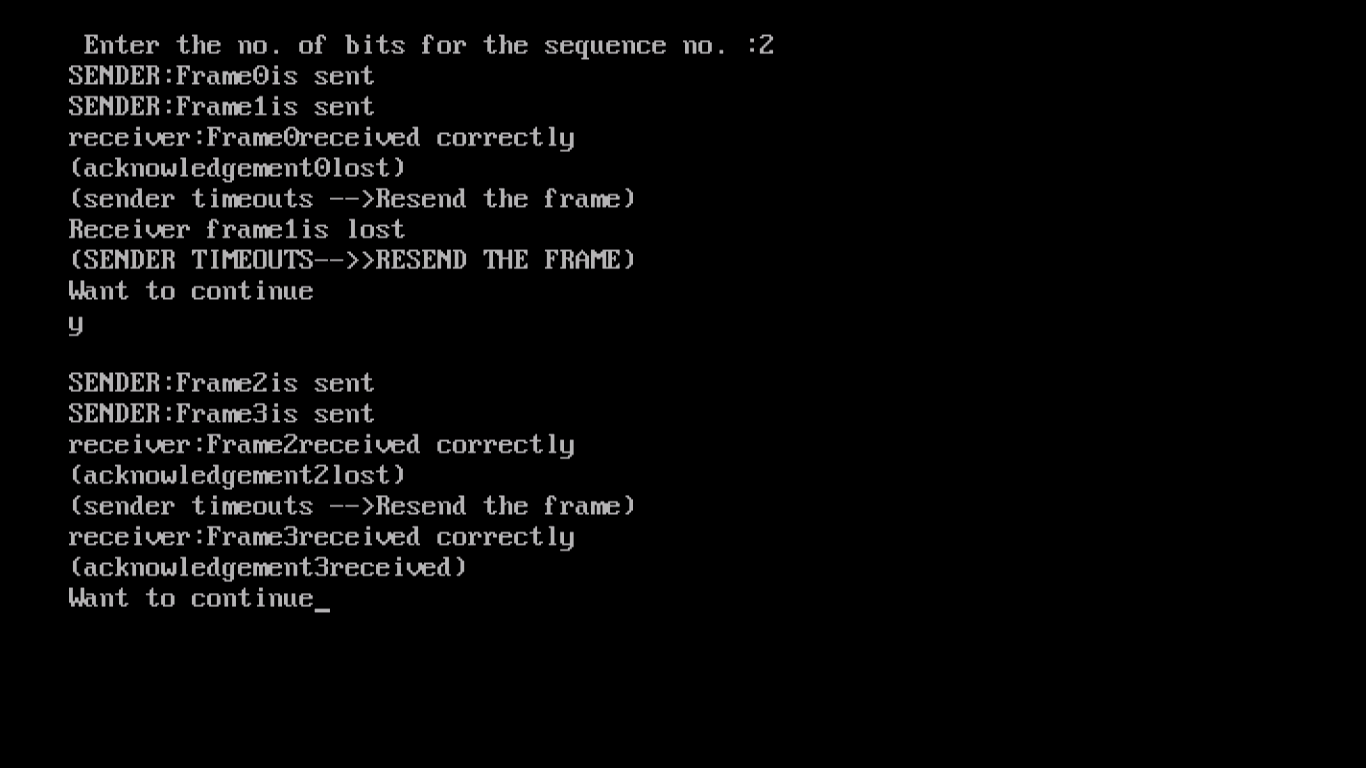
sel\_repeat sr;

sr.input();

getch();

}

**Output:-**



**Program 6**

**Write a program in C++ to generate cipher text using caeser cipher substitution technique.**

#include<stdio.h>

#include<conio.h>

#include<iostream.h>

#include<string.h>

class ct

{

char s1[20], s2[20];

int i, x, l;

public:

void getdata();

void display();

};

void ct::getdata()

{

cout<<"\n Enter the string upto 20:";

gets(s1);

int p;

cout<<"\n Enter the key";

cin>>p;

l=strlen(s1);

for(i=0;i<1;i++)

{

if((s1[i]>=97)&&(s1[i]<=122))

{

if(s1[i]==120||s1[i]==121||s1[i]==122)

{

x=s1[i]-26+p;

s2[i]=x;

}

else

{

x=s1[i]+p;

s2[i]=x;

}

}

if((s1[i]>=65)&&(s1[i]<=90))

{

if(s1[i]==88||s2[i]==89||s1[i]==90)

{

x=s1[i]-26+p;

s2[i]=x;

}

else

{

x=s1[i]+p;

s2[i]=x;

}

}

}

}

void ct::display()

{

cout<<"\n Cipher text is:";

for(i=0;i<1;i++)

{

cout<<s2[i];

}

}

void main()

{

clrscr();

ct c;

c.getdata();

c.display();

getch();

}

**Output:-**



**Program 7**

**Write a program in C++ to generate vipher text using playfair substitution technique.**

#include<stdio.h>

#include<conio.h>

#include<iostream.h>

int check(char table [5][5],char k)

{

int i,j;

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

for(j=0;j<5;++j)

{

if (table [i][j]==k)

return 0 ;

}

return 1;

}

void main()

{

clrscr();

int i,j,key\_len;

char table [5][5];

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

for(j=0;j<5;++j)

table[i][j]='0';

cout<<"\n \*\*\*\*\*\* playfair cipher \*\*\*\*\*\* \n \n";

cout<<"\n Enter the length of the key :";

cin>>key\_len;

char key[50];

cout<<"\n Enter the key :";

for(i=0;i<key\_len;i++)

{

cin>>key[i];

if (key [i]=='j')

key [i]='i';

getch();

}

int flag;

int count=0;

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

{

for(j=0;j<5;++j)

{

flag=0;

while(flag!=1)

{

if(count>key\_len)

goto l1;

flag=check(table,key[count]);

++count;

}

table[i][j]=key[(count-1)];

}

}

l1:cout<<"\n";

int val=97;

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

{

for(j=0;j<5;++j)

{

if (table[i][j]>=97&&table[i][j]<=123)

{

}

else

{

flag=0;

while(flag!=1)

{

if('j'==(char)val)

++val;

flag=check(table,(char)val);

++val;

}

table[i][j]=(char)(val-1);

}

}

}

cout<<"the table is as follows :";

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

{

for(j=0;j<5;++j)

{

cout<<table[i][j];

}

cout<<"\n";

}

int l=0;

cout<<"\n Enter the length of plain text (without spaces) :";

cin>>l;

cout<<"\n Enter the plain text :";

char p[50];

for(i=0;i<l;++i)

{

cin>>p[i];

}

for(i=0;i<l;++i)

{

if(p[i]=='j')

p[i]='i';

}

cout<<"\n The replaced text (j with i) :";

for(i=-1;i<l;++i)

cout<<p[i];

count=0;

for(i=-1;i<l;++i)

{

if(p[i]==p[i+1])

count=count+0;

}

cout<<"\n The cipher has to enter bogus char.It is either 'x' or'z'";

int length=0;

if((l+count)%2!=0)

length=(l+count);

else

length=(l+count);

cout<<"\n Value of length is \n"<<length;

char p1[50];

char temp1;

int count1=0;

for(i=-1;i<l;++i)

{

p1[count1]=p[i];

if(p[i]==p[i+1])

{

count1=count1+1;

if(p[i]=='x')

p1[count1]='z';

else

p1[count1]='x';

}

count1=count1+1;

}

char bogus;

if((1+count)%2!=0)

{

if(p1[length-1]=='x')

p1[length]='z';

else

p1[length]='x';

}

cout<<"\n The final text is :";

for(i=0;i<=length;++i)

cout<<p1[i];

char cipher\_text[50];

int r1,r2,c1,c2;

int k1;

for(k1=1;k1<=length;++k1)

{

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

{

for(j=0;j<5;++j)

{

if (table[i][j]==p1[k1])

{

r1=i;

c1=j;

}

else

if(table[i][j]==p1[k1+1])

{

r2=i;

c2=j;

}

}

}

if(r1==r2)

{

cipher\_text[k1]=table[r1][(c1+1)%5];

cipher\_text[k1+1]=table[r1][(c2+1)%5];

}

else

if(c1==c2)

{

cipher\_text[k1]=table[(r1+1)%5][c1];

cipher\_text[k1+1]=table[(r2+1)%5][c1];

}

else

{

cipher\_text[k1]=table[r1][c2];

cipher\_text[k1+1]=table[r2][c1];

}

k1=k1+1;

}

cout<<"\n \n The cipher text is :";

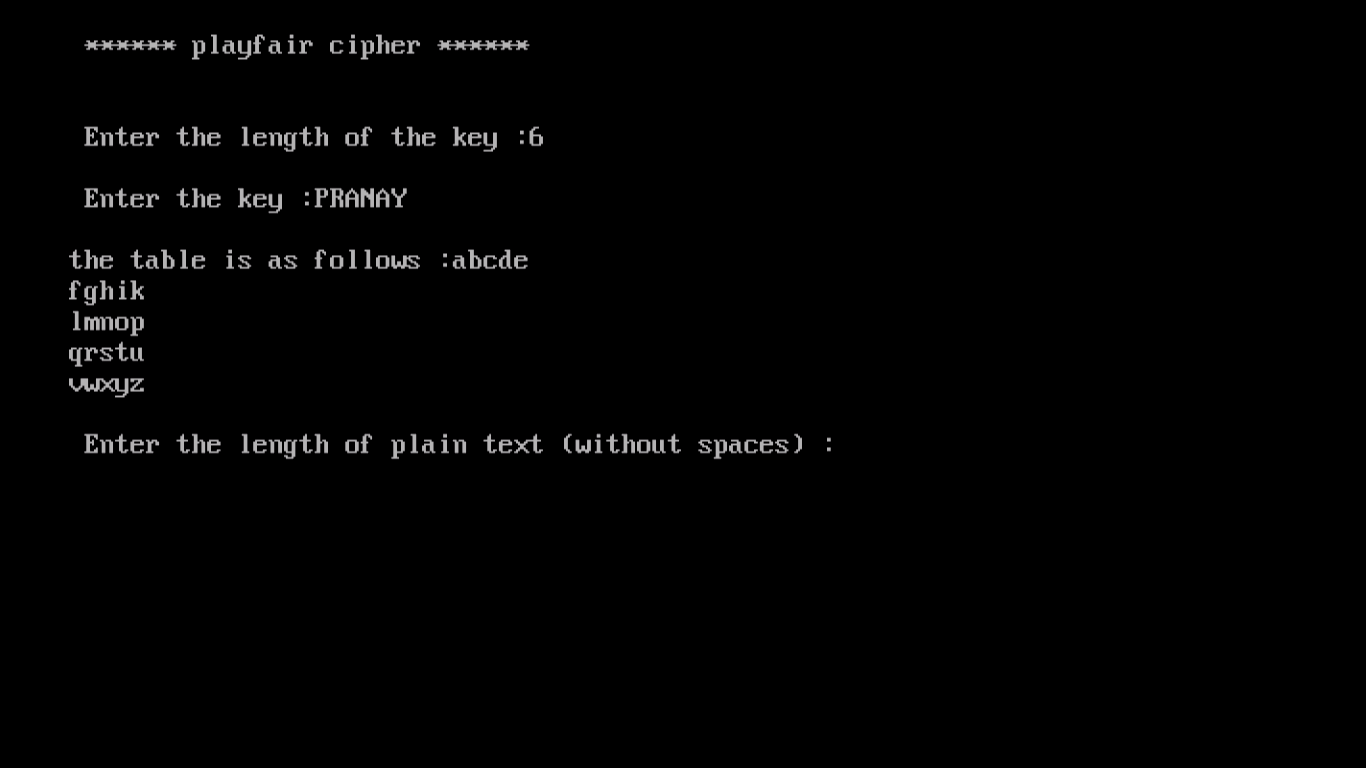
for(i=1;i<=length;++i)

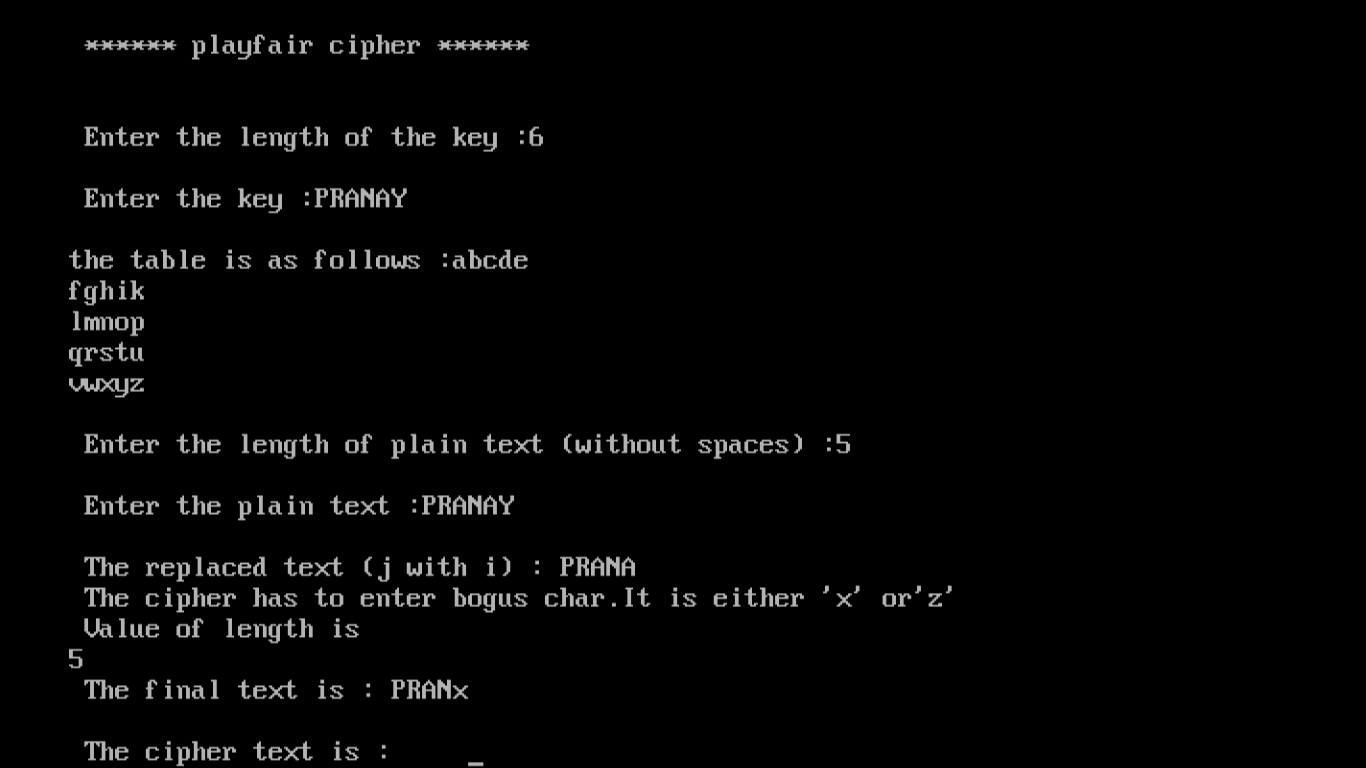
cout<<cipher\_text[i];

getch();

}

**Output:-**





**Program 8**

**Write a program in C++ to generate cipher text using transposition technique.**

#include<stdio.h>

#include<string.h>

#include<conio.h>

void cipher (int i,int c);

int findMin();

void makeArray(int,int);

char arr[22][22],darr [22][22],emessage[111],retmessage[111],key[55];

char temp[55],temp2[55];

int k=0;

int main()

{

char message[30];

int i,j,klen,emlen,flag=0;

int r,c,index,rows;

clrscr();

printf("Enter the key \n");

gets(key);

printf("Enter the message to be ciphered \n");

gets(message);

strcpy(temp,key);

klen=strlen(key);

k=0;

for(i=0;;i++)

{

if(flag==1)

break;

for(j=0;key[j]!=NULL;j++)

{

if(message[k]==NULL)

{

flag=1;

arr[i][j]='-';

}

else

{

arr[i][j]=message[k++];

}

}

}

r=i;

c=j;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

printf("%c",arr[i][j]);

}

printf("\n");

}

k=0;

for(i=0;i<klen;i++)

{

index=findMin();

cipher(index,r);

}

emessage[k]='\0';

printf("\n Encrypted message is \n");

for(i=0;emessage[i]!=NULL;i++)

printf("%c",emessage[i]);

printf("\n \n");

emlen=strlen(emessage);

strcpy(temp,key);

rows=emlen/klen;

rows;

j=0;

for(i=0,k=1;emessage[i]!=NULL;i++,k++)

{

temp2[j++]=emessage[i];

if((k%rows)==0)

{

temp2[j]='\0';

index=findMin();

makeArray(index,rows);

j=0;

}

}

printf("\n Array Retrieved is \n");

k=0;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

printf("%c",darr[i][j]);

retmessage[k++]=darr[i][j];

}

printf("\n");

}

retmessage[k]='\0';

printf("\n Mesage Retrived is \n");

for(i=0;retmessage[i]!=NULL;i++)

printf("%c",retmessage[i]);

getch();

return(0);

}

void cipher(int i,int r)

{

int j;

for(j=0;j<r;j++)

{

emessage[k++]=arr[j][i];

}

emessage[k]='\0';

}

void makeArray(int col,int row)

{

int i;

for(i=0;i<row;i++)

{

darr[i][col]=temp2[i];

}

}

int findMin()

{

int j,min,index;

min=temp[0];

index=0;

for(j=0;temp[j]!=NULL;j++)

{

if(temp[j]<min)

{

min=temp[j];

index=j;

}

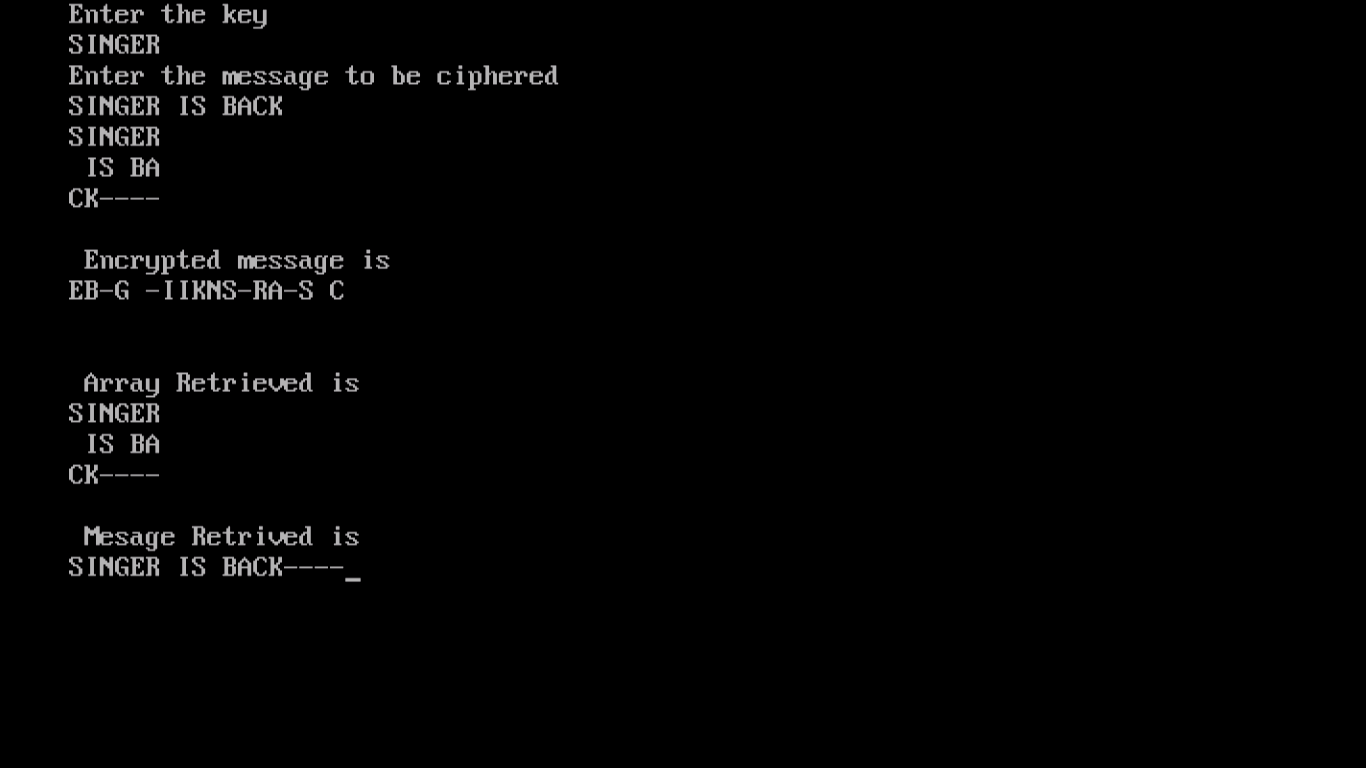
}

temp[index]=123;

return(index);

}

**Output:-**



**Program 9**

**Write a program in C++ for congestion control using leaky bucket algorithm.**

#include<iostream.h>

#include<stdlib.h>

#include<conio.h>

#include<dos.h>

#define NOF\_PACKETS 10

int rand(int a)

{

int rn=(random(a)%10)%a;

return rn==0?1:rn;

}

void main()

{

clrscr();

int packet\_sz[NOF\_PACKETS],i,clk,b\_size,o\_rate,p\_sz\_rn,p\_time,op;

for(i=-0;i<NOF\_PACKETS;++i)

packet\_sz[i]=rand(6)\*10;

for(i=0;i<NOF\_PACKETS;++i)

cout<<"\npacket["<<i<<"]:"<<packet\_sz[i]<<"bytes";

cout<<"\nenter the output rate:";

cin>>o\_rate;

cout<<"\nenter the bucket size";

cin>>b\_size;

for(i=0;i<NOF\_PACKETS;++i)

{

if((packet\_sz[i]+p\_sz\_rn)>b\_size)

if(packet\_sz[i]>b\_size)

cout<<"\n\n incoming packet size:"<<packet\_sz[i]<<"bytes is greater than bucket capacity :"<<b\_size<<"bytes\_packet rejected";

else

cout<<"\n\n bucket capacity exceeded packets rejected!!";

else

{

p\_sz\_rn+=packet\_sz[i];

cout<<"\n\n incoming packet size:"<<packet\_sz[i];

cout<<"\n bytes remaining to transmit:"<<p\_sz\_rn;

p\_time=rand(4)\*10;

cout<<"\n time left for transmition:"<<p\_time<<"units";

for(clk=10;clk<=p\_time;clk+=10)

{

sleep(1);

if(p\_sz\_rn)

{

if(p\_sz\_rn<=o\_rate)

{

op=p\_sz\_rn;

p\_sz\_rn=0;

}

else

{

op=o\_rate;

p\_sz\_rn=o\_rate;

}

cout<<"\n packet of size transmited"<<op;

cout<<"---bytes remaining to transmit:"<<p\_sz\_rn;

}

else

{

cout<<"\n time left for transmmition :"<<p\_time-clk<<"units";

cout<<"\n no packets to transmmit!!";

}

}

}

}

// getch();

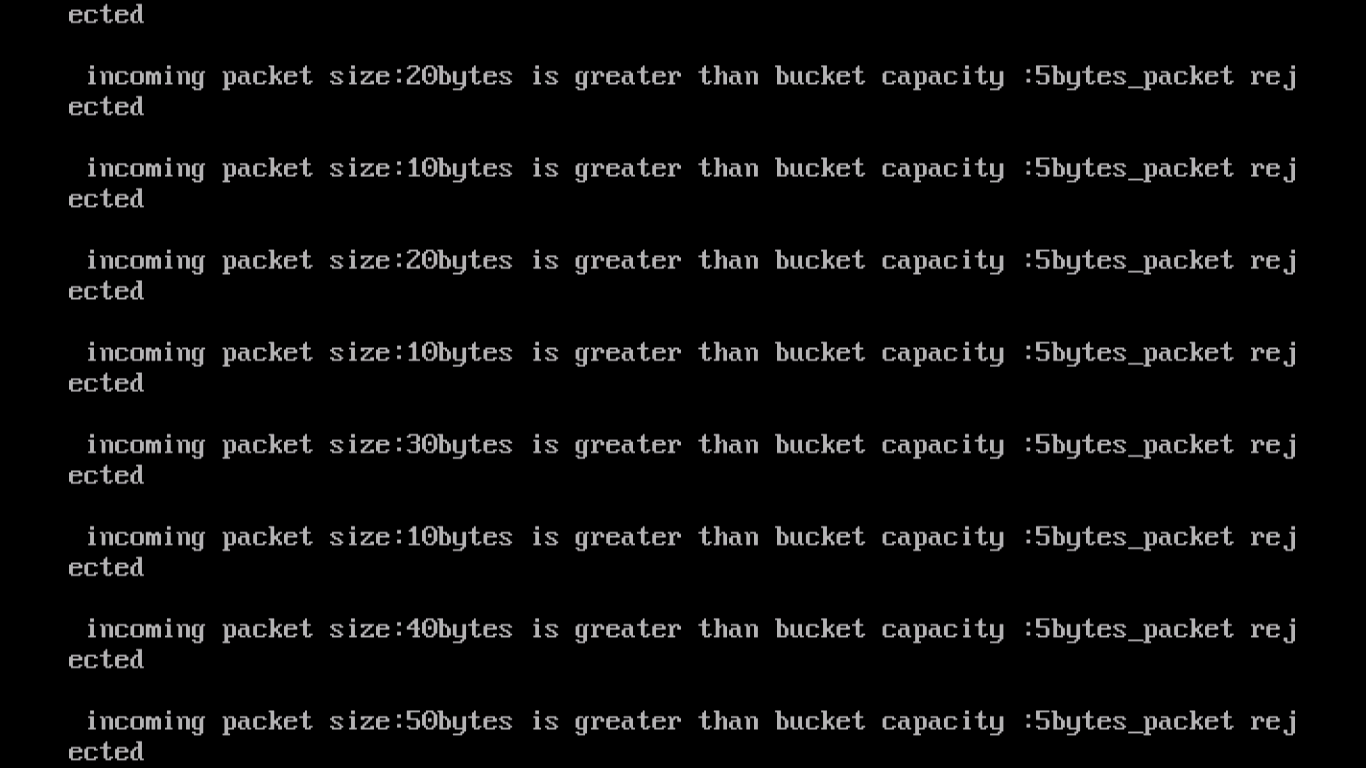
// return 1;

getch();

}

**Output:-**





**Program 10**

**Write a program in C++ to implement extended Euclidean algorithm to find the GCD of two polynomials.**

#include<iostream.h>

#include<conio.h>

class equ

{

int a1, a2, b1, b2, c1, c2;

int x, y, z;

int q, r;

public:

void getdata();

void calculate();

void display();

};

void equ::getdata()

{

cout<<"\n Enter the 1st EQU:\n";

cout<<"\n Enter the value of ax:";

cin>>a1;

cout<<"\n Enter the value of by:";

cin>>b1;

cout<<"\n Enter the value of c:";

cin>>c1;

cout<<"\n Enter the 2nd EQU:\n";

cout<<"\n Enter the value of ax:";

cin>>a2;

cout<<"\n Enetr the value of by:";

cin>>b2;

cout<<"\n Enter the value of c:";

cin>>c2;

cout<<"\n\n\n\t\t The EQU is:";

cout<<a1<<"x+"<<b1<<"y="<<c1;

cout<<"\n\n\n\t\t The EQU is:";

cout<<a2<<"x+"<<b2<<"y="<<c2;

}

void equ::calculate()

{

if(a1<=a2||a1>=a2)

{

int a3, b3, c3;

int a4, b4, c4;

a3=a2\*a1;

b3=b2\*a1;

c3=c2\*a1;

a4=a1\*a2;

b4=b1\*a2;

c4=c1\*a2;

a2=a3; b2=b3; c2=c3;

a1=a4; b1=b4; c1=c4;

}

int p;

if(a1==a2)

{

p=a1-a2;

q=b1-b2;

r=c1-c2;

}

y=r/q;

x=(c1-(b1\*y))/a1;

}

void equ::display()

{

cout<<"\n\n\t\t RESULT::::::x="<<x<<"\t y="<<y;

while(y!=0)

{

const int temp=y;

y=x%y;

x=temp;

}

if(x<=0)

{

x=x\*(-1);

}

cout<<"\n\n\n\n The GCD is that \t:::"<<x;

}

void main()

{

clrscr();

equ e;

e.getdata();

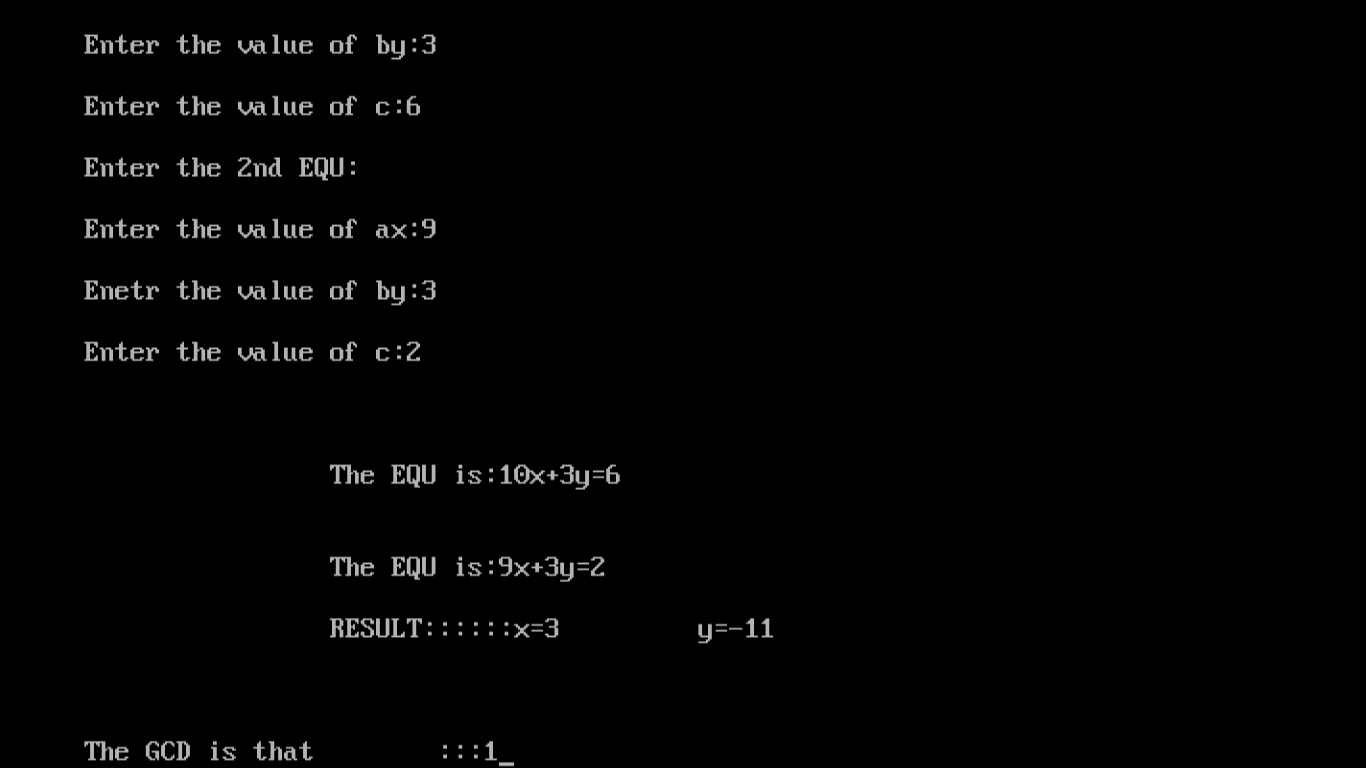
e.calculate();

e.display();

getch();

}

**Output:-**



**Program 11**

**Write a program in C++ to implement extended Euclidean algorithm to find multiplicative inverse modulo.**

#include<iostream.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

int millerrobin(int,int);

void main()

{

int num1,num2;

clrscr();

cout<<"\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n OUTPUT";

cout<<"\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n Enter number check for prime:";

cin>>num1;

cout<<"\n Enter probability parameter:";

cin>>num2;

int ans=millerrobin(num1, num2);

if(ans==1)

cout<<"\n Number is prime:";

else

cout<<"\n Number is not prime:";

getch();

}

int millerrobin(int n, int a1)

{

int q, m, y, a, z, i;

double lo;

q=n-1;

lo=log(n);

int p=a1\*lo;

for(i=1; i<p;i++)

{

m=q;

y=1;

int ra=random(n);

a=ra%q+1;

z=a;

while(m>0)

{

while(m%2==0)

{

int x=z;

z=(z\*z)%m;

if((z==1)&&(x!=q))

return(0);

m=m/2;

}

m=m-1;

y=(y\*z)%n;

}

if(y!=1)

{

return(0);

}

}

return(1);

}

**Output:-**



**Program 12**

**Write a program in C++ to RSA algorithm.**

#include<math.h>

#include<iostream.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

class RSA

{

long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;

char msg[100];

public:

void getdata();

int prime(long int);

void ce();

long int cd(long int);

void encrypt();

void decrypt();

};

void RSA::getdata()

{

clrscr();

cout<<"\nENTER THE FIRST PRIME NUMBER \n";

cin>>p;

cout<<"\nENTER THE ANOTHER PRIME NUMBER \n";

cin>>q;

cout<<"\nENTER MESSAGE \n";

cin>>msg;

for(i=0; msg[i]!=NULL;i++)

m[1]=msg[i];

n=p\*q;

t=(p-1)\*(q-1);

ce();

}

void RSA::ce()

{

int k;

k=0;

for(i=2;i<t;i++)

{

if(t%i==0)

continue;

flag=prime(i);

if(flag==1 && i!=p && i!=q)

{

e[k]=i;

flag=cd(e[k]);

if(flag>0)

{

d[k]=flag;

k++;

}

d[k]=flag;

k++;

}

if(k==99)

break;

}

}

long int RSA::cd(long int x)

{

long int k=1;

while(1)

{

k=k+t;

if(k%x==0)

return(k/x);

}

}

void RSA::encrypt()

{

long int pt,ct,key=e[0],k,len;

i=0;

len=strlen(msg);

while(i!=len)

{

pt=m[i];

pt=pt-96;

k=1;

for(j=0;j<key;j++)

{

k=k\*pt;

k=k%n;

}

temp[i]=k;

ct=k+96;

en[i]=ct;

i++;

}

en[i]=-1;

cout<<"\nTHE ENCRYPTED MESSAGE IS \n";

for(i=0;en[i]!=-1;i++)

printf("%c",en[i]);

}

void RSA::decrypt()

{

long int pt,ct,key=d[0],k;

i=0;

while(en[i]!=-1)

{

ct=temp[i];

k=1;

for(j=0;j<key;j++)

{

k=k\*ct;

k=k%n;

}

pt=k+96;

m[i]=pt;

i++;

}

m[i]=-1;

cout<<"\nTHE DECRYPTED MESSAGE IS\n";

for(i=0;m[i]!=-1;i++)

printf("%c", m[i]);

}

int RSA::prime(long int pr)

{

int i,j;

j=sqrt(pr);

for(i=2;i<=j;i++)

{

if(pr%i==0)

return 0;

}

return 1;

}

void main()

{

clrscr();

RSA r1;

r1.getdata();

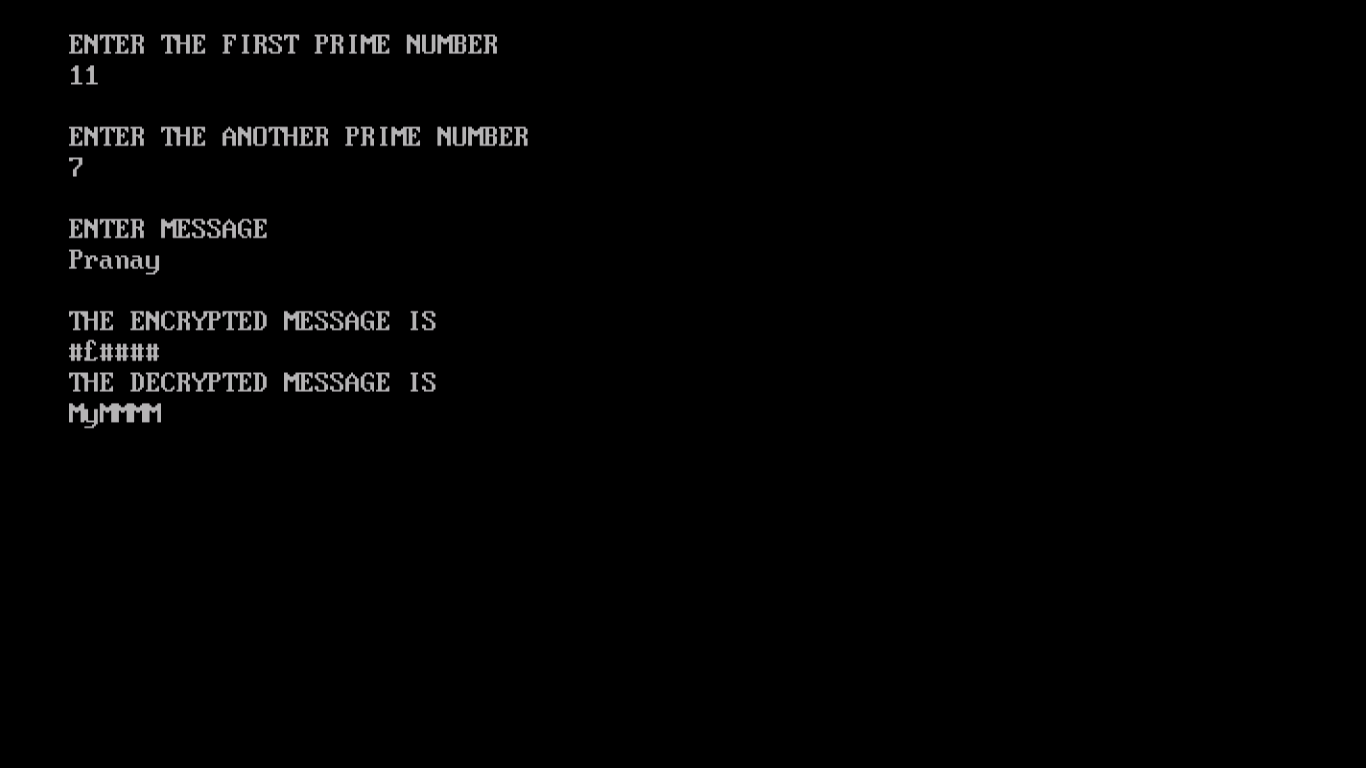
r1.encrypt();

r1.decrypt();

getch();

}

**Output:-**



**Program 14**