

ERROR DETECTION AND CORRECTION

1. C program to implement Hamming code

SOURCE CODE:

```
#include<stdio.h>
```

```
#include<math.h>
```

```
int n;
```

```
int data2[100];
```

```
int calc(int k)
```

```
{int r,z;
```

```
for(r=0;;r++)
```

```
{if(pow(2,r)>=k+r+1)
```

```
    {break;}
```

```
else
```

```
{continue;}}
```

```
z= k+r;
```

```
return z;}
```

```
void codegen(int data[],int num1 )
```

```
{int i,l,j=0,k=0,x=0,y=0;
```

```
for(i=0;i<num1;i++)
```

```
{ if(i!=((int)pow(2,k))-1)
```

```
    {data2[i]=data[j];
```

```
    j++;}
```

```
else
```

```
{k++;;}}
```

```
{for(i=0;i<num1;i++)
{if(i==(pow(2,y))-1)
{if(num1==11)
{ if(i==0)
{data2[0]=data2[2]^data2[4]^data2[6]^data2[8]^data2[10];}
else if(i==1)
{data2[1]=data2[2]^data2[5]^data2[6]^data2[9]^data2[10];}
else if(i==3)
{data2[3]=data2[4]^data2[5]^data2[6];}
else if(i==7)
{data2[7]=data2[8]^data2[9]^data2[10];}}
else if(num1==7)
{if(i==0)
{data2[0]=data2[2]^data2[4]^data2[6];}
else if(i==1)
{data2[1]=data2[2]^data2[5]^data2[6];}
else if(i==3)
{data2[3]=data2[4]^data2[5]^data2[6];}
}

y++;}
else
{x++;
}}}
```

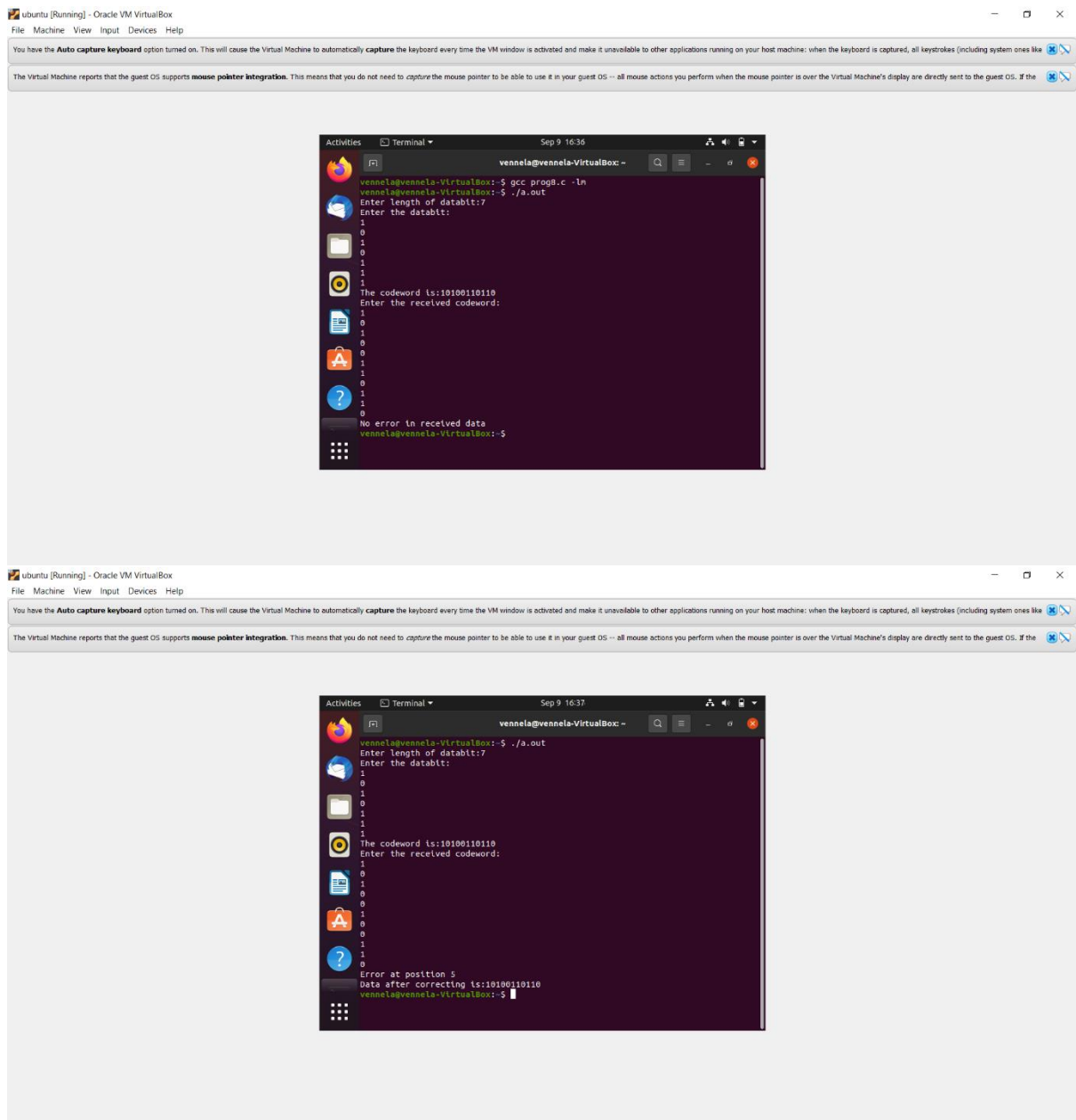
```
void check(int recdata[],int n)
```

```
{int s1,s2,s3,s4,c,i;
if(n>=8)
{
s1=recdata[0]^recdata[2]^recdata[4]^recdata[6]^recdata[8]^recdata[10];
s2=recdata[1]^recdata[2]^recdata[5]^recdata[6]^recdata[9]^recdata[10];
s3=recdata[3]^recdata[4]^recdata[5]^recdata[6];
s4=recdata[7]^recdata[8]^recdata[9]^recdata[10];
c=(8*s4)+(4*s3)+(2*s2)+(1*s1);}
else if(n<8)
{
s1=recdata[0]^recdata[2]^recdata[4]^recdata[6];
s2=recdata[1]^recdata[2]^recdata[5]^recdata[6];
s3=recdata[3]^recdata[4]^recdata[5]^recdata[6];

c=(4*s3)+(2*s2)+(1*s1);
}
if(c==0)
{printf("No error in received data\n");
}
else
{printf("Error at position %d \n",c);
if(recdata[c-1]==0)
{recdata[c-1]=1;}
else
{recdata[c-1]=0;}
printf("Data after correcting is:");
for(i=n-1;i>=0;i--)
{printf("%d",recdata[i]);
}
}
```

```
printf("\n");  
}}  
  
int main()  
{int k,i,j,n;int data[100];int recdata[100];  
printf("Enter length of databit:");  
scanf("%d",&k);  
printf("Enter the databit:\n");  
for(i=k-1;i>=0;i--)  
{scanf("%d",&data[i]);  
}  
n=calc(k);  
codegen(data,n);  
printf("The codeword is:");  
for(i=n-1;i>=0;i--)  
{printf("%d",data2[i]);}  
printf("\n");  
printf("Enter the received codeword:\n");  
  
for(i=n-1;i>=0;i--)  
{scanf("%d",&recdata[i]);  
}  
check(recdata,n);  
return 0;}
```

OUTPUT:



```
vennela@vennela-VirtualBox:~$ gcc prog8.c -ln
vennela@vennela-VirtualBox:~$ ./a.out
Enter length of databit:7
Enter the databit:
1
0
1
0
1
1
1
The codeword is:10100110110
Enter the received codeword:
1
0
1
0
0
1
1
0
1
1
0
No error in received data
vennela@vennela-VirtualBox:~$
```

```
vennela@vennela-VirtualBox:~$ ./a.out
Enter length of databit:7
Enter the databit:
1
0
1
0
1
1
1
The codeword is:10100110110
Enter the received codeword:
1
0
1
0
0
1
1
0
1
1
0
Error at position 5
Data after correcting is:10100110110
vennela@vennela-VirtualBox:~$
```

2. C program to implement Checksum

SOURCE CODE:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
char data[100];
```

```
int rightSum(int l)
```

```
{
```

```
    int sum=0, i=1;
```

```
    for(;i<l;i=i+2)
```

```
        sum=sum + (int) data[i];
```

```
    return sum;
```

```
}
```

```
int leftSum(int l)
```

```
{
```

```
    int sum=0, i=0;
```

```
    for(;i<l;i=i+2)
```

```
        sum=sum + (int) data[i];
```

```
    return sum;
```

```
}
```

```
int main()
```

```
{
```

```
    char buf[100];
```

```
    int i, n, op=0, irs=0, ils=0, prs=0, cls=0, wc=0, pls=0, s=0, ocs=0, len=0;
```

```
    while(op==0)
```

```
    {
```

```
        printf("*** Checksum Program ***\n\n1. Sender\n2. Receiver\n3. Exit\nEnter your choice...");
```

```
        scanf("%d",&n);
```

```
        switch(n)
```

```
        {
```

case 1:

```
{  
    printf("\nEnter the data to be transmitted -> ");  
    fgets(buf,100,stdin);  
    scanf("%[^\n]s",data);  
    len=strlen(data);  
    if(len%2!=0)  
        len++;  
    irs=rightSum(len);    //initial right sum  
    prs=irs%256;          //partial right sum  
    cls=irs/256;          //carry to left sum  
    ils=cls+leftSum(len); //initial left sum  
    pls=ils%256;          //partial left sum  
    wc=ils/256;           //Wrapping carry  
    s=pls*256+prs+wc;  
    ocs = 65535 - s;  
    printf("The checksum generated is %X\n", ocs);  
}
```

break;

case 2:

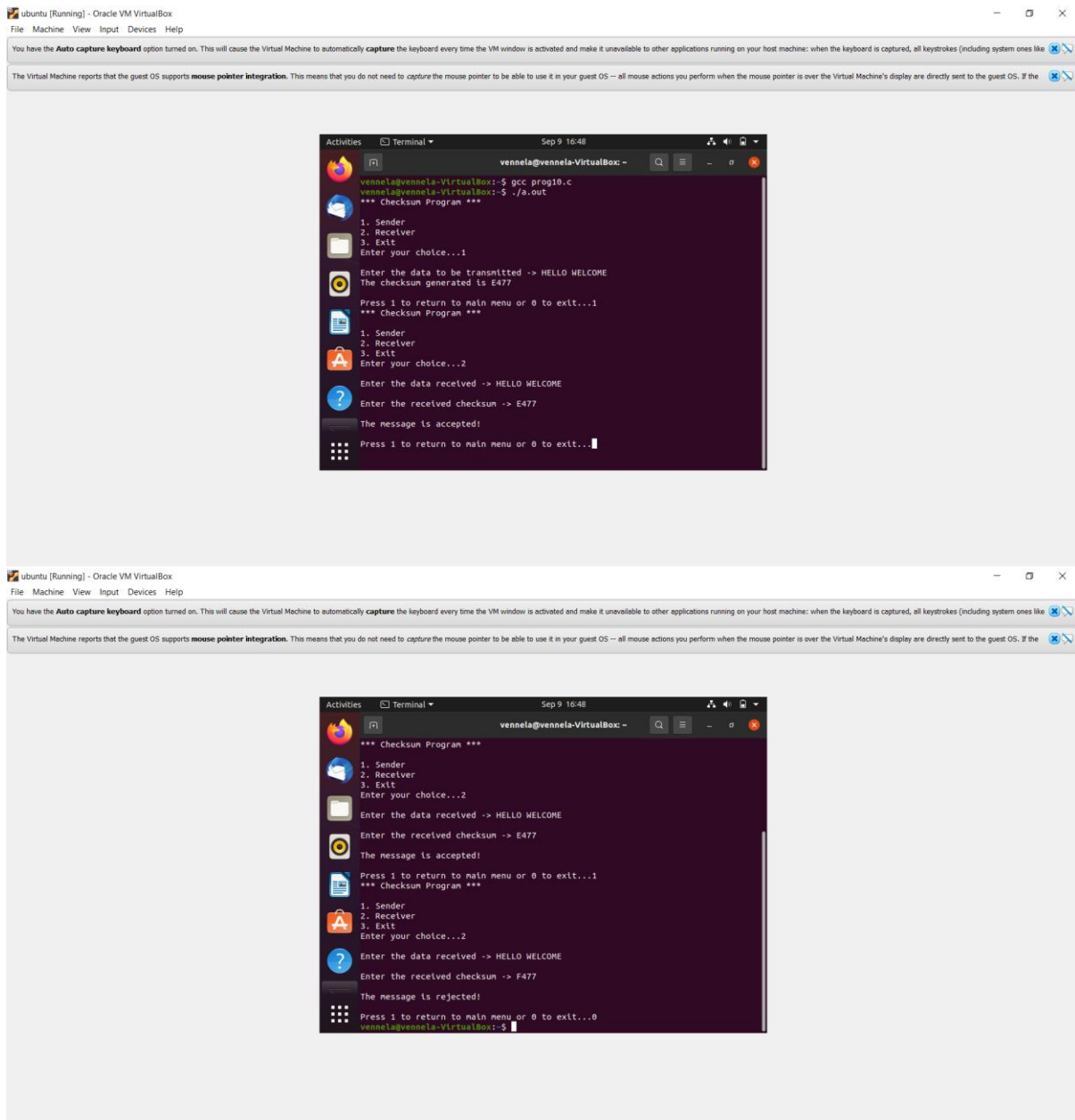
```
{  
    char cs[100];  
    int ch[100];  
    printf("\nEnter the data received -> ");  
  
    fgets(buf,100,stdin);  
    scanf("%[^\n]s",data);  
    printf("\nEnter the received checksum -> ");  
    fgets(buf,100,stdin);
```

```
scanf("%[^\\n]s",cs);
len=strlen(data);
if(len%2!=0)
len++;
for(i=0;i<strlen(cs);i++)
{
    if(cs[i]>='0' && cs[i]<='9')
        ch[i]=cs[i]-48;
    else if(cs[i]>='A' && cs[i]<='F')
        ch[i]=cs[i]-55;
    else if(cs[i]>='a' && cs[i]<='f')
        ch[i]=cs[i]-87;
}
irs=rightSum(len) + ch[2]*16 + ch[3];    //initial right sum
prs=irs%256;        //partial right sum
cls=irs/256;        //carry to left sum
ils=cls+leftSum(len) + ch[0]*16 + ch[1]; //initial left sum
pls=ils%256;        //partial left sum
wc=ils/256;         //Wrapping carry
s=pls*256+prs+wc;
ocs = 65535 - s;
if(ocs==0)
printf("\\n\\nThe message is accepted!\\n");
else
printf("\\n\\nThe message is rejected!\\n");
}
break;
case 3: exit(0);
}
```



```
printf("\nPress 1 to return to main menu or 0 to exit...");  
  
scanf("%d", &i);  
  
if(i==0)  
  
op=1;  
  
}  
  
return 0;}
```

OUTPUT:



3. C program to implement Cyclic Redundancy code(CRC)

SOURCE CODE:

```
#include <stdio.h>

#include <string.h>

void main() {

    int i,j,keylen,msglen,z=0,y=0;

    char input[100],
key[30],rem1[30],key2[30],temp[30],quot[100],quot1[100],temp1[30],rem[30],key1[30],buf
[100],reccode[100];


    printf("Enter Data: ");
    scanf("%[^\n]s",input);

    printf("Enter Key: ");
    fgets(buf,100,stdin);
    scanf("%[^\n]s",key);


    keylen=strlen(key);


    msglen=strlen(input);
    strcpy(key1,key);
    for (i=0;i<keylen-1;i++) {
        input[msglen+i]='0';
    }
    for (i=0;i<keylen;i++)
        temp[i]=input[i];
    for (i=0;i<msglen;i++) {
        quot[i]=temp[0];
        if(quot[i]=='0')
```

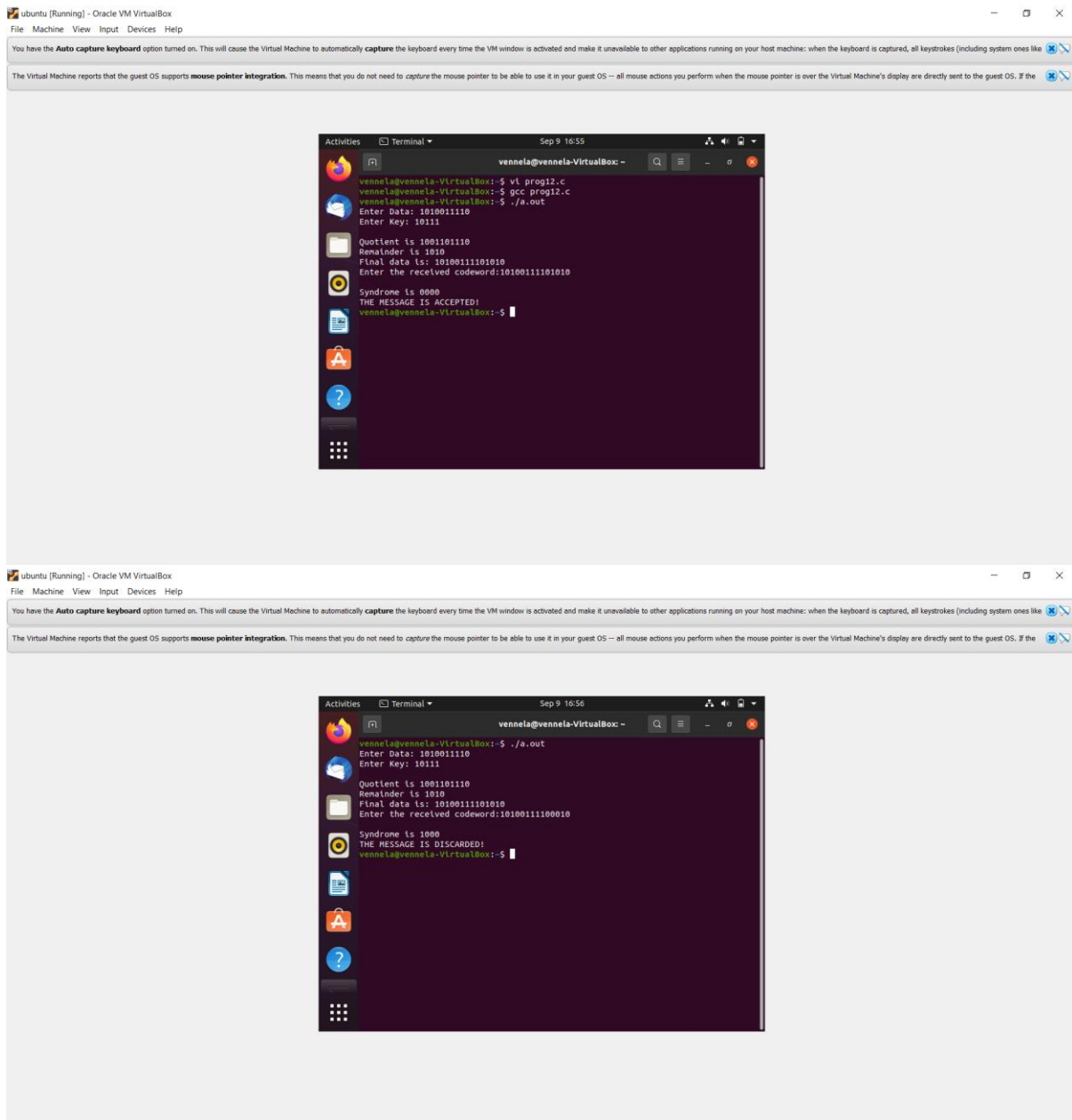
```
    for (j=0;j<keylen;j++)
        key[j]='0';
    else
        for (j=0;j<keylen;j++)
            key[j]=key1[j];
    for (j=keylen-1;j>0;j--)
        if(temp[j]==key[j])
            rem[j-1]='0'; else
            rem[j-1]='1';

    rem[keylen-1]=input[i+keylen];
    strcpy(temp,rem);
}
strcpy(rem,temp);
printf("\nQuotient is ");
for (i=0;i<msglen;i++)
    printf("%c",quot[i]);
printf("\nRemainder is ");
for (i=0;i<keylen-1;i++)
    printf("%c",rem[i]);
printf("\nFinal data is: ");
for (i=0;i<msglen;i++)
    printf("%c",input[i]);
for (i=0;i<keylen-1;i++)
    printf("%c",rem[i]);
printf("\n");
printf("Enter the received codeword:");
fgets(buf,100,stdin);
scanf("%[^\n]s",reccode);
```

```
strcpy(key2,key1);
for (i=0;i<keylen;i++)
    temp1[i]=reccode[i];
for (i=0;i<msglen;i++) {
    quot1[i]=temp1[0];
    if(quot1[i]=='0')
        for (j=0;j<keylen;j++)
            key1[j]='0'; else
            for (j=0;j<keylen;j++)
                key1[j]=key2[j];
    for (j=keylen-1;j>0;j--) {
        if(temp1[j]==key1[j])
            rem1[j-1]='0'; else
            rem1[j-1]='1';
    }
    rem1[keylen-1]=reccode[i+keylen];
    strcpy(temp1,rem1);
}
strcpy(rem1,temp1);
printf("\nSyndrome is ");
for (i=0;i<keylen-1;i++)
    {printf("%c",rem1[i]);}
printf("\n");
for (i=0;i<keylen-1;i++)
    {
        if(rem1[i]=='0' || rem1[i]=='\0')
            {z=1;}
        else if(rem1[i]!='0')
            {y=1;}}
```

```
if(z==1 && y==0 )  
  
    {printf("THE MESSAGE IS ACCEPTED!\n");}  
  
else  
  
    {printf("THE MESSAGE IS DISCARDED!\n");}
```

OUTPUT:



The image displays two screenshots of a terminal window running a program. The terminal is titled 'vennela@vennela-VirtualBox' and shows the following sequence of events:

- The user enters the date: 1010011110
- The user enters the key: 10111
- The program calculates and displays: Quotient is 1001101110, Remainder is 1010, Final date is: 10100111101010
- The user enters the received codeword: 10100111101010
- The program displays: Syndrome is 0000
- In the first screenshot, the program displays: THE MESSAGE IS ACCEPTED!
- In the second screenshot, the program displays: THE MESSAGE IS DISCARDED!