LAB 6 - **VOIP PACKET LOSS RECOVERY TECHNIQUES**

DHARMITHA S SHETTY

200948030

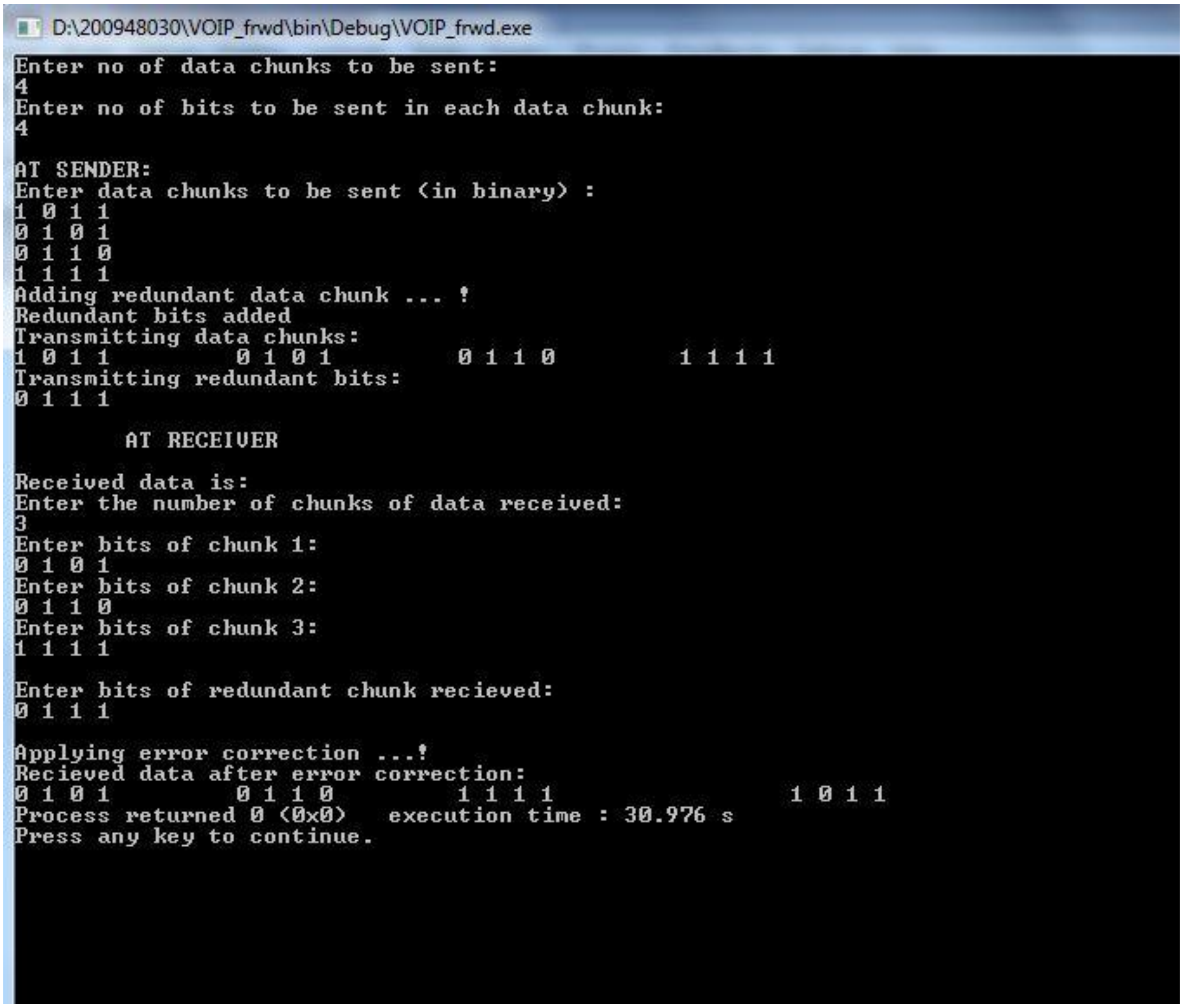
Simple Forward Error Correction (FEC)

C Program:

```
#include <stdio.h>
int main()
{
   int n,b,n1;
   int i,j;
   int a[20][20], r[20][20];
   printf("Enter no of data chunks to be sent:\n");
   scanf("%d",&n);
   printf("Enter no of bits to be sent in each data chunk:\n");
   scanf("%d",&b);

   printf("\nAT SENDER:\n");
   printf("Enter data chunks to be sent (in binary) :\n");
   for(i=0;i<n;i++)
      for(j=0;j<b;j++)
         scanf("%d",&a[i][j]);
   printf("Adding redundant data chunk ... !\n");
   printf("Redundant bits added\n");
   printf("Transmitting data chunks:\n");
   for(i=0;i<n;i++)
   {
      for(j=0;j<b;j++)
         printf("%d ",a[i][j]);
      printf("\t");
   }
   printf("\n");
   printf("Transmitting redundant bits:\n");
   for(i=0;i<n-1;i++)
      for(j=0;j<b;j++)
         if(a[i][j] == a[i+1][j])
            a[i+1][j] = 0;
         else
            a[i+1][j] = 1;

   for(i=0;i<n;i++)
      for(j=0;j<b;j++)
         if(i==n-1)
            printf("%d ",a[i][j]);
   printf("\t");
```

```
    printf("\n\n \tAT RECEIVER\n\n");
    printf("Received data is:\n");
    printf("Enter the number of chunks of data received:\n");
    scanf("%d",&n1);
    if(n1!=n-1)
    {
        printf("Lost packets cannot be reconstructed\n");
        exit(0);
    }
    for(i=0;i<n1;i++)
    {
        printf("Enter bits of chunk %d:\n",i+1);
        for(j=0;j<b;j++)
            scanf("%d",&a[i][j]);
    }
    printf("\nEnter bits of redundant chunk recieved:\n");
    for(j=0;j<b;j++)
        printf("%d ",a[n-1][j]);
    printf("\n\nApplying error correction ...!\n");
    printf("Recieved data after error correction:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i<n-1)
                printf("%d ",a[i][j]);
        }
        printf("\t");
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<b;j++)
        {
            r[i][j] = a[i][j];
            a[i+1][j] = a[i+1][j]^a[i][j];
            if(i==n-1)
                printf("%d ",r[i][j]);
        }
    }
    return 0;
}
```

OUTPUT:



```
Enter no of data chunks to be sent:
4
Enter no of bits to be sent in each data chunk:
4

AT SENDER:
Enter data chunks to be sent (in binary) :
1 0 1 1
0 1 0 1
0 1 1 0
1 1 1 1
Adding redundant data chunk ... !
Redundant bits added
Transmitting data chunks:
1 0 1 1         0 1 0 1         0 1 1 0         1 1 1 1
Transmitting redundant bits:
0 1 1 1

        AT RECEIVER

Received data is:
Enter the number of chunks of data received:
3
Enter bits of chunk 1:
0 1 0 1
Enter bits of chunk 2:
0 1 1 0
Enter bits of chunk 3:
1 1 1 1

Enter bits of redundant chunk recieved:
0 1 1 1

Applying error correction ...!
Recieved data after error correction:
0 1 0 1         0 1 1 0         1 1 1 1                 1 0 1 1
Process returned 0 (0x0)   execution time : 30.976 s
Press any key to continue.
```

2.Interleaving

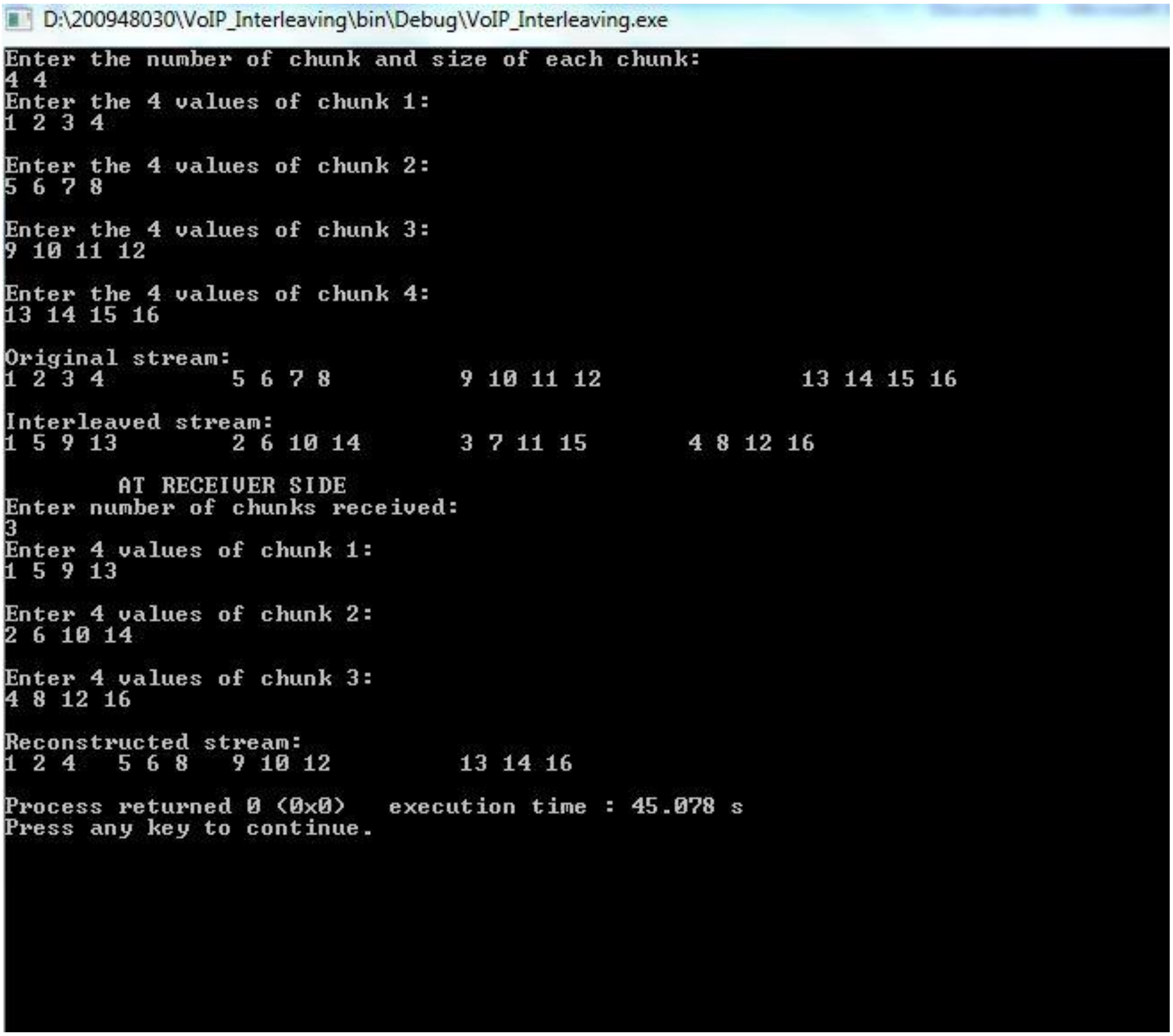
C PROGRAM :

```
#include <stdio.h>
int main()
{
   int n,b,size,n1;
   int i,j;
   int a[20];
   printf("Enter the number of chunk and size of each chunk: \n");
   scanf("%d%d",&n,&b);
   size = n*b;
   for(i=0;i<n;i++)
   {
```

```
   printf("Enter the %d values of chunk %d:\n",b,i+1);
   for(j=(i*b);j<(i*b)+b;j++)
   {
      scanf("%d",&a[j]);
   }
   printf("\n");
}

printf("Original stream:\n");
for(i=0;i<size;i++)
{
   if((i+1)%b==0)
      printf("%d\t\t",a[i]);
   else
      printf("%d ",a[i]);
}
printf("\n");

printf("\nInterleaved stream:\n");
for(i=0;i<b;i++)
{
   for(j=0;j<n;j++)
   {
      printf("%d ",a[i+(j*b)]);
   }
   printf("\t");
}
printf("\n");

printf("\n\tAT RECEIVER SIDE\n");

printf("Enter number of chunks received:\n");
scanf("%d",&n1);

for(i=0;i<n1;i++)
{
   printf("Enter %d values of chunk %d:\n",b,i+1);
   for(j=(i*n);j<(i*n)+n;j++)
   {
      scanf("%d",&a[j]);
   }
   printf("\n");
}
printf("Reconstructed stream:\n");
```

```
   for(i=0;i<n;i++)
   {
      for(j=0;j<n1;j++)
      {
         printf("%d ",a[i+(j*n)]);
      }
      printf("\t");
   }
   printf("\n");
   return 0;
}
```

OUTPUT:



D:\200948030\VoIP_Interleaving\bin\Debug\VoIP_Interleaving.exe

```
Enter the number of chunk and size of each chunk:
4 4
Enter the 4 values of chunk 1:
1 2 3 4

Enter the 4 values of chunk 2:
5 6 7 8

Enter the 4 values of chunk 3:
9 10 11 12

Enter the 4 values of chunk 4:
13 14 15 16

Original stream:
1 2 3 4         5 6 7 8         9 10 11 12              13 14 15 16

Interleaved stream:
1 5 9 13        2 6 10 14       3 7 11 15       4 8 12 16

        AT RECEIVER SIDE
Enter number of chunks received:
3
Enter 4 values of chunk 1:
1 5 9 13

Enter 4 values of chunk 2:
2 6 10 14

Enter 4 values of chunk 3:
4 8 12 16

Reconstructed stream:
1 2 4   5 6 8   9 10 12         13 14 16

Process returned 0 (0x0)   execution time : 45.078 s
Press any key to continue.
```