CSE1004 LAB3

BY ASHUTOSH ARDU
REGISTRATION NUMBER- 20BRS1262

EVEN AND ODD PARITY

ALGORITHM

ODD PARITY

IF THE NUMBER OF 1's IN THE GIVEN DATAWORD IS EVEN THEM ADD:
1 AS PARITY BIT TO THE DATAWORD
ELSE ADD
0 AS PARITY BIT TO THE DATAWORD

EVEN PARITY

IF THE NUMBER OF 1's IN THE GIVEN DATAWORD IS ODD THEM ADD:
1 AS PARITY BIT TO THE DATAWORD
ELSE ADD
0 AS PARITY BIT TO THE DATAWORD

CODE:

```
#include<stdio.h>
#include<string.h>
// Parity check

int even(int num)
{
   int count=0,k;
   while(num>0)
   {
        k=num%10;
        num/=10;
        if(k==1)
            ++count;
   }
   if(count%2==0)
        return 0;
```

```
        else
            return 1;
    }

    int odd(int num)
    {
        int count=0,k;
        while(num>0)
        {
            k=num%10;
            num/=10;
            if(k==1)
                ++count;
        }
        if(count%2==0)
            return 1;
        else
            return 0;
    }

    int main()
    {
        int data,l,out[3],parity;
        scanf("%d",&data);
        printf("Which parity 1_Even or 2_Odd\n");
        scanf("%d",&l);
        if(l==1)
            parity=even(data);
        else if(l==2)
            parity=odd(data);
        out[0]=data;out[1]=parity;
        printf("Your encoded data %d%d",out[0],out[1]);
    }
```

OUTPUT:

EVEN PARITY

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

cd "/home/ghost/Desktop/C-C++/Error_task/" && gcc Parity.c -o Parity && "/home/ghost/Desktop/C-C++/Error_task/"Parity
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/Error_task/" && gcc Parity.c -o Parity && "/home/ghost/Desktop/C-C++/Error_task/"Parity
10110
Which parity 1_Even or 2_Odd
1
Your encoded data 101101

┌──(ghost㉿kali)-[~/Desktop/C-C++/Error_task]
└─$
```

ODD PARITY

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

└─$ cd "/home/ghost/Desktop/C-C++/Error_task/" && gcc Parity.c -o Parity && "/home/ghost/Desktop/C-C++/Error_task/"Parity
10110
Which parity 1_Even or 2_Odd
2
Your encoded data 101100

┌──(ghost㉿kali)-[~/Desktop/C-C++/Error_task]
└─$
```

CHECKSUM:

ALGORITHM:

AT THE SENDER END:

- FIRST INPUT ALL THE SEGMENTS AND THEN ADD THEN AMONG EACHOTHER USING 1's COMPLEMENT ADDITION.

- THE COMPLEMENT THE OBTAINED SUM IS THE REQUIRED CHECKSUM.

- SEND THE WHOLE DATAWORD ALONG WITH ITS CHECKSUM.

AT THE RECEIVER'S END:

- FIRST RECIEVE THE ALL THE DATA SEGMENTS ALONG WITH THEIR CHECKSUM.

- THEN ADD ALL THE DATAWORD ALONG WITH THEIR CHECKSUM USING 1's COMPLEMENT ADDITION.

- IF THE COMPLEMENT OF THE SUM IS 00000000 THEN THE DATA IS WELL RECEIVED, ELSE THE DATA IS CORRUPTED.

CODE:

PLEASE NOTE : THE BELOW GIVEN CODE IS WRITTEN PARTLY BY ME AND PARTLY IS REFFERED FROM TCP/IP BOOK (FROM CHECKSUM AT THE APPENDIX SECTION AND FROM Implementation of CHECKSUM | Basic , medium ,expert programs example in c,java,c/++ (scanftree.com) FOR IMPLEMENTATION OF SPECIAL BINARY ADDITION)

```c
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

int* decToBin(int dec,int n)// FOR CONVERTION CARRY FROM INT TO
{                           // ITS BINARY FORM
    int * bin=(int *)calloc(n,sizeof(int));
    for(int i=n-1;i>=0;i--)
```

```
    {
        bin[i]=dec%2;
        dec/=2;
    }
    return bin;
}

void checksumGen(int **data,int n,int k)
{
    int carrynum=0;
    for(int j=k-1;j>=0;j--)
    {
        int sum=0;
        for(int i=0;i<n;i++)
        {
            sum+=data[i][j];
        }
        sum+=carrynum;
        data[n][j]=sum%2;
        carrynum=sum>0 ? sum>>1 : 0;
    }
    int *carry=decToBin(carrynum,k);

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<k;j++)
        {
            printf("%d ",data[i][j]);
        }
        printf("<-Segment [%d] \n",(i+1));
    }
    printf("-----------------\n");
    for(int j=0;j<k;j++)
        printf("%d ",data[n][j]);
    printf("<-Sum1\n");
    carrynum=0;
    for(int i=k-1;i>=0;i--)
    {
        int sum=carrynum+carry[i]+data[n][i];
        data[n][i]=sum%2;
        carrynum=sum>0 ? sum>>1 : 0;
    }
    for(int i=0;i<k;i++)
        printf("%d ",carry[i]);
    printf("<-Carry\n");
    printf("-----------------\n");
```

```
        for(int i=0;i<k;i++)
        {
            printf("%d ",data[n][i]);
            data[n][i]= data[n][i]==0 ? 1 : 0;
        }
        printf("<-Sum2\n");
        printf("-----------------\n");
        for(int i=0;i<k;i++)
        {
            printf("%d ",data[n][i]);
        }
        printf("<-CHECKSUM\n");
    }

    void checksumChk(int **data,int n,int k)
    {
        int *chkBucket=(int *)calloc(k,sizeof(int));
        int carrynum=0;
        for(int j=k-1;j>=0;j--)
        {
            int sum=0;
            for(int i=0;i<=n;i++)
            {
                sum+=data[i][j];
            }
            sum+=carrynum;
            chkBucket[j]=sum%2;
            carrynum=sum>0 ? sum>>1 : 0;
        }
        int *carry=decToBin(carrynum,k);

        for(int i=0;i<n;i++)
        {
            for(int j=0;j<k;j++)
            {
                printf("%d ",data[i][j]);
            }
            printf("<-Segment [%d] \n",(i+1));
        }
        for(int i=0;i<k;i++)
        {
            printf("%d ",data[n][i]);
        }
        printf("<-CHECKSUM (Receiver)\n");
        printf("-----------------\n");
        for(int j=0;j<k;j++)
            printf("%d ",chkBucket[j]);
        printf("<-Sum1\n");
```

```
        carrynum=0;
        for(int i=k-1;i>=0;i--)
        {
            int sum=carrynum+carry[i]+chkBucket[i];
            chkBucket[i]=sum%2;
            carrynum=sum>0 ? sum>>1 : 0;
        }
        for(int i=0;i<k;i++)
            printf("%d ",carry[i]);
        printf("<-Carry\n");
        printf("-----------------\n");
        for(int i=0;i<k;i++)
        {
            printf("%d ",chkBucket[i]);
            chkBucket[i]=chkBucket[i]==0 ? 1 : 0;
        }
        printf("<-Sum2\n");
        printf("-----------------\n");
        bool accept=true;
        for(int i=0;i<k;i++)
        {
            printf("%d ",chkBucket[i]);
            if(chkBucket[i]!=0)
                accept=false;
        }
        printf("<-CHECKSUM\n");
        printf("%s",(accept ? "Accepted!" : "Rejected!"));
        printf("\n");
    }

    int main()
    {
        int n,k;
        printf("Enter no of Segments: ");
        scanf("%d",&n);
        printf("Enter bit length of each segment: ");
        scanf("%d",&k);
        int len=(n+1)*sizeof(int *)+ (n+1)*(k)*sizeof(int);
        int **data=(int **)malloc(len);
        int * ptr=(int *)(data+n+1);
        for(int i=0;i<n+1;i++)
            data[i]=(ptr+k*i);
        for(int i=0;i<n;i++)
        {
            printf("Enter segment[%d] (space separated): ",(i+1));
            for(int j=0;j<k;j++)
            {
```

```
                scanf("%d",&data[i][j]);
            }
        }
        checksumGen(data,n,k);
        printf("\nNow sender is sending the segments with checksum ...
\n\n");
        printf("----------------------------------------\n");
        printf("Transmission Successful\nAt the receiver end...\n");
        checksumChk(data,n,k);
    }
```

OUTPUT:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
└─$ cd "/home/ghost/Desktop/C-C++/Error_task/" && gcc checksum.c -o checksum && "/home/ghost/Desktop/C-C++/Error_task/"checksum
Enter no of Segments: 3
Enter bit length of each segment: 8
Enter segment[1] (space separated): 1 1 1 1 0 1 1 1
Enter segment[2] (space separated): 1 1 0 0 1 0 1 0
Enter segment[3] (space separated): 1 0 0 1 0 0 0 0
1 1 1 1 0 1 1 1 <-Segment [1]
1 1 0 0 1 0 1 0 <-Segment [2]
1 0 0 1 0 0 0 0 <-Segment [3]
-----------------
0 1 0 1 0 0 0 1 <-Sum1
0 0 0 0 0 0 1 0 <-Carry
-----------------
0 1 0 1 0 0 1 1 <-Sum2
-----------------
1 0 1 0 1 1 0 0 <-CHECKSUM

Now sender is sending the segments with checksum ...

----------------------------------------
Transmission Successful
At the receiver end...
1 1 1 1 0 1 1 1 <-Segment [1]
1 1 0 0 1 0 1 0 <-Segment [2]
1 0 0 1 0 0 0 0 <-Segment [3]
1 0 1 0 1 1 0 0 <-CHECKSUM (Receiver)
-----------------
1 1 1 1 1 1 0 1 <-Sum1
0 0 0 0 0 0 1 0 <-Carry
-----------------
1 1 1 1 1 1 1 1 <-Sum2
-----------------
0 0 0 0 0 0 0 0 <-CHECKSUM
Accepted!

┌──(ghost㉿kali)-[~/Desktop/C-C++/Error_task]
└─$
```

CRC MECHANISM:

ALGORITHM:

- FIRST RECEIVE THE DATAWORD AND THE DIVISOR FROM THE POLYNOMIAL GENERATOR.

-THEN ADD (NUMBER OF BITS IN DIVISOR - 1) NUMBER OF ZERO TO THE DATAWORD.

-THEN DIVIDE THE DATAWORD WITH THE DIVISOR USING BINARY DIVISION.

- THE REMAINDER OBTAINED IS THE CRC PARITY BITS AND IS REPLACED WITH THE EARLIER ADDED EXTRA ZEROS IN THE DATAWORD.

-THIS NEWLY UPDATED DATAWORD IS THEN SENT TO THE RECEIVER.

-THE RECEIVER RECEIVES THE DATAWORD ALONG WITH THE DIVISOR.

-THE RECEIVER THEN DIVIDES THE DATAWORD WITH DIVISOR USING BINARY DIVISION.

-IF THE REMAINDER IS EQUAL (NUMBER OF BITS IN DIVISOR - 1) NUMBER OF BITS OF ZERO THEN THE DATA RECEIVED IS RIGHT.

CODE:

```
#include<stdio.h>
//#include<conio.h>
#include<string.h>
void main() {
   int i,j,keylen,msglen;
   char input[100], key[30],temp[30],quot[100],rem[30],key1[30];
   //clrscr();
   printf("Enter Data: ");
   scanf("%s",input);
   printf("Enter Key: ");
   scanf("%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]);

    char temp1[30],rem1[30],quot1[100],error=0;
    char newdata[100];
     printf("\nReceiver end\nEnter the data recieved\n");
     scanf("%s",newdata);

     keylen=strlen(key);
     msglen=strlen(newdata);
        strcpy(key1,key);
    for (i=0;i<keylen-1;i++) {
        newdata[msglen+i]='0';
    }
    for (i=0;i<keylen;i++)
     temp1[i]=newdata[i];
    for (i=0;i<msglen;i++) {
        quot1[i]=temp1[0];
        if(quot1[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(temp1[j]==key[j])
             rem1[j-1]='0'; else
             rem1[j-1]='1';
        }
        rem1[keylen-1]=newdata[i+keylen];
        strcpy(temp1,rem1);
    }
    strcpy(rem1,temp1);

    for(i=0;i<keylen-1;++i)
        if(rem1[i]=='1')
        {
```

```
                error=1;
                break;
            }
        if(error==1)
            printf("\nError in the data recieved\n");
        else
            printf("\nData well recieved");
        //getch();
    }
```

OUTPUT:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

└─$ cd "/home/ghost/Desktop/C-C++/Error_task/" && gcc crc.c -o crc && "/home/ghost/Desktop/C-C++/Error_task/"crc
Enter Data: 101100101
Enter Key: 11110

Quotient is 111001011
Remainder is 0010
Final data is: 1011001010010
Reciever end
Enter the data recieved
1011001010010

Data well recieved

┌──(ghost㉿kali)-[~/Desktop/C-C++/Error_task]
└─$
```