Computer Networking

PCA-2

Name: Soumyajit Saha

Year: 3rd(6th Semester)

Class Roll No.: CSE2018/005

University Roll No.: 11700118037

Paper Code: PCC-CS692

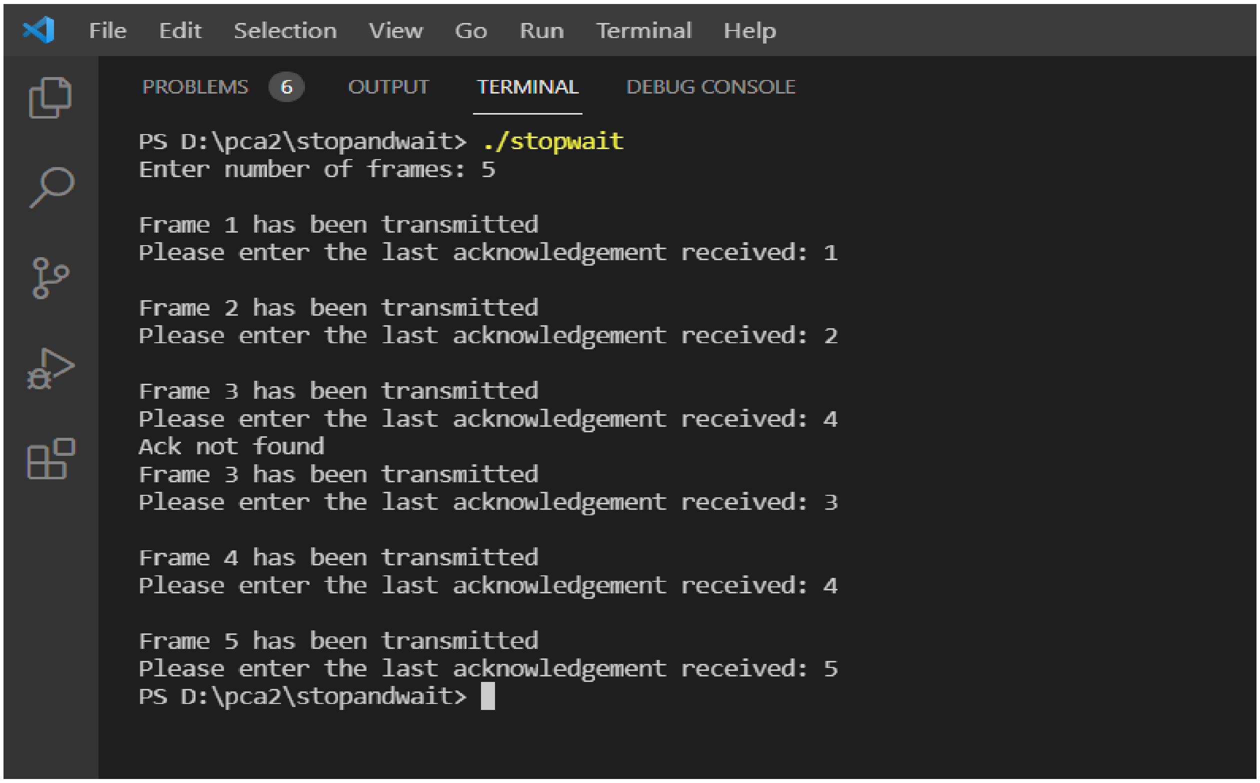
Q1. Write a code to implement stop and wait protocol using C.

Source Code:

```
#include<stdio.h>
int main()
{
    int framesize,sent=0,ack;
    printf("Enter number of frames: ");
    scanf("%d",&framesize);
    while(sent<=framesize-1)
    {
        printf("\nFrame %d has been transmitted",sent+1);
        printf("\nPlease enter the last acknowledgement received: ");
        scanf("%d",&ack);
        //sent++;
        if((sent+1)!=ack){
            printf("Ack not found");
            continue;
        }
        ++sent;
    }

  return 0;
}
```

Output:



```
PS D:\pca2\stopandwait> ./stopwait
Enter number of frames: 5

Frame 1 has been transmitted
Please enter the last acknowledgement received: 1

Frame 2 has been transmitted
Please enter the last acknowledgement received: 2

Frame 3 has been transmitted
Please enter the last acknowledgement received: 4
Ack not found
Frame 3 has been transmitted
Please enter the last acknowledgement received: 3

Frame 4 has been transmitted
Please enter the last acknowledgement received: 4

Frame 5 has been transmitted
Please enter the last acknowledgement received: 5
PS D:\pca2\stopandwait>
```

Q2. Write a code on Error detection technique Cyclic redundancy Check (CRC) in C? (Data word : 100100 Key : 1101)

Source Code:

Sender:

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>


int main() {

        printf("Enter the size of the message : ");

        int n;

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

```
printf("Enter the size of the diivisor : ");

int m;

scanf("%d",&m);

int mess[n + m -1]; //initializing the array for message

int d[m]; //initializing the array for the divisor


printf("Enter the data of the message : \n");


for(int i=0;i<n;i++)

        scanf("%d",&mess[i]);


printf("Enter the divisor : \n");


for(int i=0;i<m ; i++)

        scanf("%d",&d[i]);


//Append 0's to the end of the message


for(int i=n;i<(n+m) ; i++)

        mess[i] = 0;


/*

for(int i=0;i<(n+m-1);i++)

        printf("%d",mess[i]);

*/


//Perform binary division on the message using the divisor
```

```
int select = m;

int temp[m];


for(int i=0;i<select;i++)

        temp[i] = mess[i];


while(select < n+m) {


        if(temp[0] == 1) {


                //Performing the xor operation


                for(int i=1;i<select;i++) {

                        if(temp[i] == d[i])

                                temp[i] = 0;

                        else

                                temp[i] = 1;

                }


                //shifting the bits left


                for(int i=0;i<m-1;i++)

                        temp[i] = temp[i+1];




                temp[m-1] = mess[select];
```

```
	}

	else {
		for(int i=0;i<m-1;i++)
			temp[i] = temp[i+1];
		temp[m-1] = mess[select];
	}

	select++;
}
printf("Resulting CRC : \n");

for(int i=0;i<m-1;i++)
	printf("%d",temp[i]);

printf("\nFinal message : ");

int k=0;

for(int i=n;i<(n+m-1);i++) {
	mess[i] = temp[k];
	k++;
}

for(int i=0;i<(n+m-1);i++) {
	printf("%d",mess[i]);
}
```

```
          return 0;
}
```

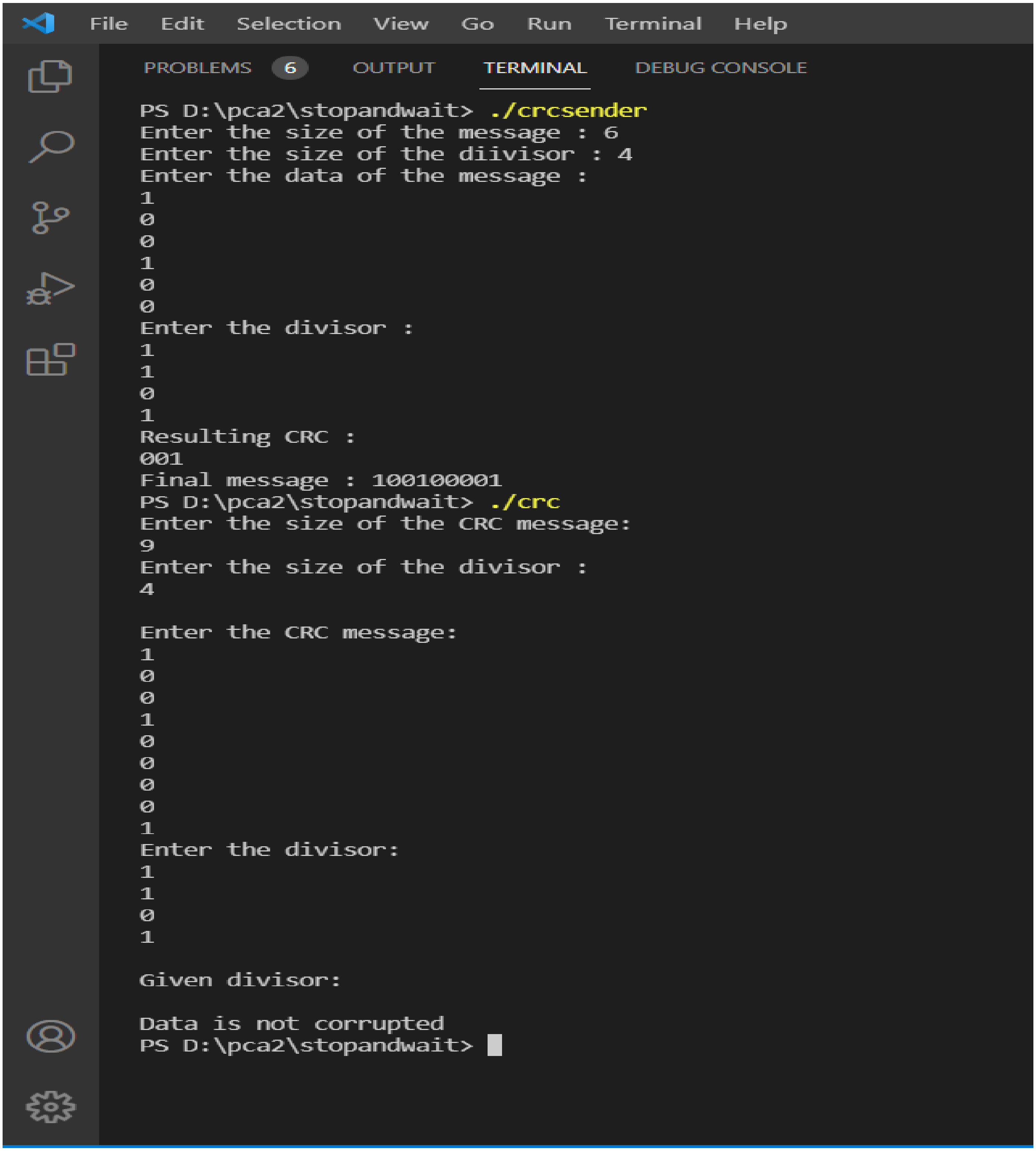
Receiver:

```
#include<stdbool.h>
#include <stdlib.h>
#include<stdio.h>
int main()
{
   printf("Enter the size of the CRC message: \n");
   int n;
   scanf("%d", &n);
   int *mess = (int *)malloc(sizeof(int)*n);
   printf("Enter the size of the divisor : \n");
   int m;
   scanf("%d", &m);
   int *d = (int *)malloc(sizeof(int)*m);
   int *d1 = (int *)malloc(sizeof(int)*m);
   printf("\nEnter the CRC message: \n");
   for(int i=0;i<n;i++)
   scanf("%d", &mess[i]);
   printf("Enter the divisor: \n");
   for(int i=0;i<m;i++)
   {
      scanf("%d", &d[i]);
```

```
    d1[i] = d[i];
}
int select = m;
int *temp = (int *)malloc(sizeof(int)*m);
for(int i=0;i<m;i++)
temp[i] = mess[i];
while(select < n)
{
    if(temp[0] == 1)
    {
        for(int i=1;i<select;i++)
        {
            if(temp[i] == d[i])
            temp[i] = 0;
            else
            temp[i] = 1;
        }
        for(int i=0;i<m-1;i++)
        temp[i] = temp[i+1];
        temp[m-1] = mess[select];
    }
    else
    {
        for(int i=0;i<m-1;i++)
        temp[i] = temp[i+1];
        temp[m-1] = mess[select];
    }
```

```
        select++;
    }
    bool flag = false;
    printf("\nGiven divisor: \n");
    for(int i=0;i<m;i++)
    {
        if(temp[i] != d[i])
        {
            flag = true;
            break;
        }
    }
    if(flag == true)
    printf("\nData corrupted \n");
    else
    printf("\nData is not corrupted \n");
    return 0;
}
```

Output:



```
PS D:\pca2\stopandwait> ./crcsender
Enter the size of the message : 6
Enter the size of the diivisor : 4
Enter the data of the message :
1
0
0
1
0
0
Enter the divisor :
1
1
0
1
Resulting CRC :
001
Final message : 100100001
PS D:\pca2\stopandwait> ./crc
Enter the size of the CRC message:
9
Enter the size of the divisor :
4

Enter the CRC message:
1
0
0
1
0
0
0
0
1
Enter the divisor:
1
1
0
1

Given divisor:

Data is not corrupted
PS D:\pca2\stopandwait>
```