Comprehensive Definition:

"Design computer controlled Traffic Signal System using circular queue data structure. In this we can set/modify timers for different operations Stop, Ready, Go. Set proper sequence of operations."

Program file:

Name of file: Traffic Signal.c

File purpose:

In this file the main function handles the whole process to manipulate the Signal Timer as we have in our real life. While using "time.h" library function I manipulated the operation they are "Stop", "Ready", "Go". The backend responsible to store the information here was "Circular Queue". Thus finally using this functions it worked properly.

Code:

```
//Traffic Signal
#include <stdio.h>
#include<time.h>
#define n 4//limiting the signals

void Sloting_Traffic(int var);//enque operation of data
void Timers_Set();//for reseting the timer
int Traffic_Que[n];//forming a queue to maintain the data
int rear=-1, front=-1;//pointers for pointing the queue
int ch,count=0;//counter to maintain a function
int timer;//setting timer for operations
```

```
void display();//displays the current state of signals


void delay(int number_of_seconds)
{
    int milli_seconds = 1000 * number_of_seconds;//storing value

    clock_t start_time = clock();//using inbuilt function

    while (clock() < start_time + milli_seconds)//manipulating data
        ;
}// to maintain the time lapse hence create a delay
void main()
{
    int x,data[2][4],flag=0;
    int temp,temp1;
    printf("-------------------------------------------------------
-------------------------------------------------------------------
-----------------\n");
    printf(" \t\t\t\t\t\t  WELCOME TO TRAFFIC SIGNAL SYSTEM \n");
    printf("-------------------------------------------------------
-------------------------------------------------------------------
-----------------\n");
    printf(" \t\t\t\t\t\tTHE MOST EASIEST WAY TO HANDLE TRAFFIC
\n");
    printf("-------------------------------------------------------
-------------------------------------------------------------------
-----------------\n\n");

    printf("\n\n************************ Enter the no. of Vehicles
for Specific Signals to Work Accordingly
****************************\n");

    for(int i=0; i<n; i++)
    {
        printf("\nSignal %d: ",i+1);
        scanf("%d",&x);
        data[0][i]=i+1;
        data[1][i]=x;
    }//scanning no. of vehicles so priority can be arranged
accordingly

    for(int i=0; i<=n-2; i++) //Selection Sorting
    {
        for(int j=i+1; j<=n-1; j++)
        {
            if(data[1][i]<data[1][j])//comparing the conditions for 
priority
            {
                temp=data[1][i];//for priority sorting by storing 
into a temporary variable
                data[1][i]=data[1][j];
                data[1][j]=temp;

                //repeating the same steps for character data
                temp1=data[0][i];//for character data sorting by 
storing into a temporary variable
                data[0][i]=data[0][j];
                data[0][j]=temp1;

            }
        }
    }

    printf("\n\n********************* Sorted Traffic 
****************\n");
    printf("\n Signal  Vehicles");
    for(int i=0; i<n; i++)
    {
        printf("\n %d  ->   %d ",data[0][i],data[1][i]);
    }//printing sorted data with vehicles

    for(int i=0; i<n; i++)
    {
        Sloting_Traffic(data[0][i]);
    }//enquing data into the queue

    printf("\n\n******************* Initial Scenario 
***************\n");
    display();//printing current scenario

    Timers_Set();//scanning timer i.e. lapse
    delay(timer);//delaying till traffic releases
    int q1=Decreasing_Traffic();//dequeing queue
    Sloting_Traffic(q1);//enqueing data

    for(int i=0;; i++)
    {
        printf("\n1. Check Status ");
        printf("\n2. Modify Timer ");
        printf("\n3. Timer Remains Same ");
        printf("\n4. Exit ");
        printf("\n\nPlease Select Your Choice: ");
        scanf("%d",&ch);//scanning choice
        if(ch==1)
        {
            printf("\n***************** Current Status for Signals
******************* \n");
            display();//displaying current status

        }
        else if(ch==2)
        {
            Timers_Set();
            delay(timer);
            q1=Decreasing_Traffic();
            Sloting_Traffic(q1);//again repeating the same
procedure

        }
        else if(ch==3)
        {
            delay(timer);
            q1=Decreasing_Traffic();
            Sloting_Traffic(q1);//similarly repeating the process
of enque and deque

        }
        else if(ch==4)
        {
            printf("\n**************** END *****************\n");
            exit(0);
            break;
        }
        else
        {
            printf("\nInvalid Choice");
        }

    }

}
```

```
void Sloting_Traffic(int var)//Enque process
{
    if((front == 0 && rear == n-1) || (front == rear+1))
    {
        //here it won't enter ever
    }
    if (front == -1 && rear==-1)//when it points to the last element of array
    {
        front = 0;//it fixes again to the first
        rear = 0;
    }
    else
    {
        if(rear == n-1)
            rear = 0;
        else
            rear = rear+1;//normal when data is entered
    }
    Traffic_Que[rear] = var;//filling data into the queue
}

int Decreasing_Traffic()//Deque procedure
{
    if(front==-1&&rear==-1)
    {
        return 0;//it won't enter here ever
    }

    else if(front==rear)
    {
        front=rear=-1;//relocating pointers
        return 0;
    }

    else
    {
        int x= Traffic_Que[front];//storing present data of the specific point
        front=(front+1)%n;//taking care of circular nature
        return x;//returning data
    }
}
```

```c
void Timers_Set()
{
    if(count==0)//when it is called for the first time
    {
        printf("\nPlease Set Your Initial Timer: ");
        scanf("%d",&timer);
        count++;
    }
    else
    {
        printf("\nPlease Enter Your Modified Timer: ");
        scanf("%d",&timer);
    }

}

void display()
{
    int arr[4],k=0;//maintaining an array to keep the sequence of
present states
    int fp = front,rp = rear;//pointing to rear and front
    if(front == -1)
    {
        printf("Queue is empty\n");//won't enter ever
        return;
    }

    if( fp <= rp )//when rear is ahead than front
        while(fp <= rp)
        {
            arr[k]=Traffic_Que[fp];//storing data into the array
            fp++;
            k++;
        }
    else
    {
        while(fp <= n-1)//when front is less than the last element
        {
            arr[k]=Traffic_Que[fp];//storing data into the array
            fp++;
            k++;
        }
        fp = 0;
        while(fp <= rp)//when it is ahead
        {
            arr[k]=Traffic_Que[fp];//storing data into the array
            fp++;
            k++;
        }
    }

    printf("\n\nSignal %d -> \"Go\"",arr[0]);
    printf("\n\nSignal %d -> \"Ready\"",arr[1]);
    printf("\n\nSignal %d -> \"Stop\"",arr[2]);
    printf("\n\nSignal %d -> \"Stop\"",arr[3]);
    printf("\n");//printing current status of signals
}
```

INPUT/OUTPUT:

```
----------------------------------------------------------------------------------------------------------------
-------------------------------------------
                                        WELCOME TO TRAFFIC SIGNAL SYSTEM
----------------------------------------------------------------------------------------------------------------
-------------------------------------------
                                        THE MOST EASIEST WAY TO HANDLE
TRAFFIC
----------------------------------------------------------------------------------------------------------------
-------------------------------------------
```

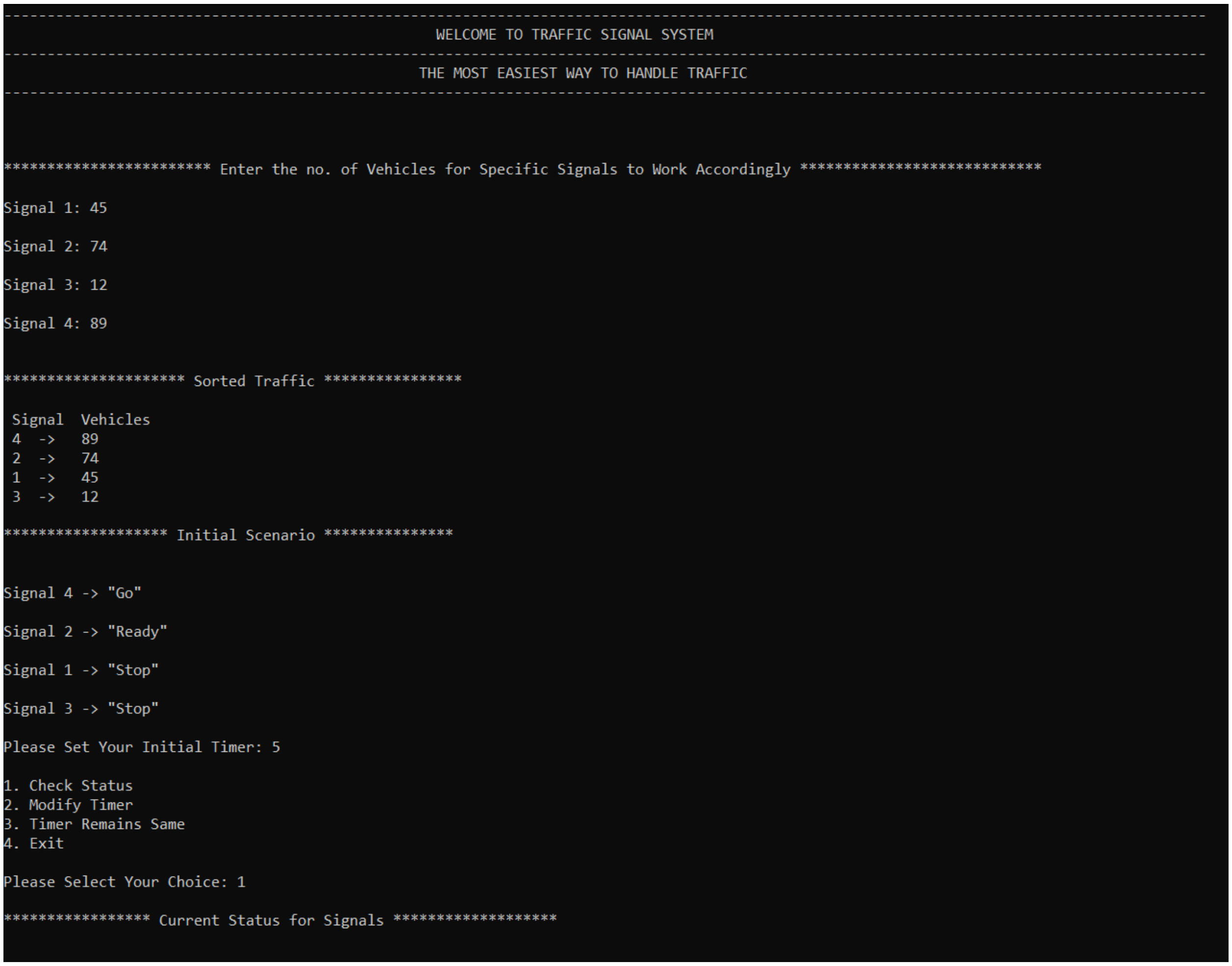
```
************************ Enter the no. of Vehicles for Specific Signals to Work Accordingly ****************************

Signal 1: 45

Signal 2: 74

Signal 3: 12

Signal 4: 89



********************* Sorted Traffic ****************

 Signal  Vehicles
 4  ->   89
 2  ->   74
 1  ->   45
 3  ->   12

******************* Initial Scenario ***************
```

```
Signal 4 -> "Go"

Signal 2 -> "Ready"

Signal 1 -> "Stop"

Signal 3 -> "Stop"

Please Set Your Initial Timer: 5


1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit


Please Select Your Choice: 1


***************** Current Status for Signals *******************
```

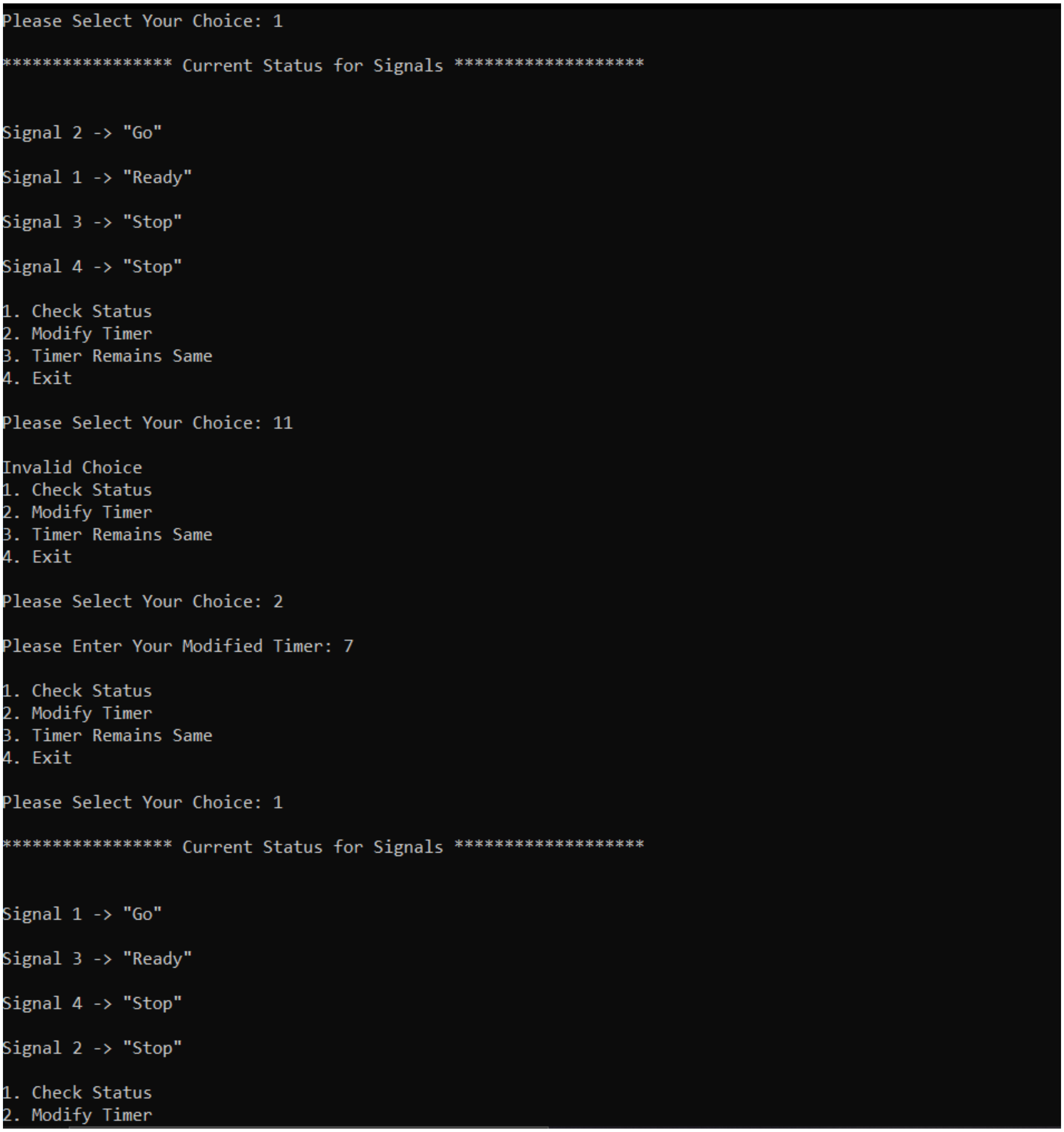
```
Signal 2 -> "Go"

Signal 1 -> "Ready"

Signal 3 -> "Stop"

Signal 4 -> "Stop"

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 11

Invalid Choice
1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit
```

```
Please Select Your Choice: 2

Please Enter Your Modified Timer: 7

1. Check Status

2. Modify Timer

3. Timer Remains Same

4. Exit

Please Select Your Choice: 1

***************** Current Status for Signals *******************

Signal 1 -> "Go"

Signal 3 -> "Ready"

Signal 4 -> "Stop"
Signal 2 -> "Stop"


1. Check Status

2. Modify Timer

3. Timer Remains Same

4. Exit


Please Select Your Choice: 3


1. Check Status

2. Modify Timer

3. Timer Remains Same

4. Exit


Please Select Your Choice: 1


***************** Current Status for Signals *******************



Signal 3 -> "Go"
Signal 4 -> "Ready"


Signal 2 -> "Stop"


Signal 1 -> "Stop"


1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit


Please Select Your Choice: 4


**************** END *****************


*/
```

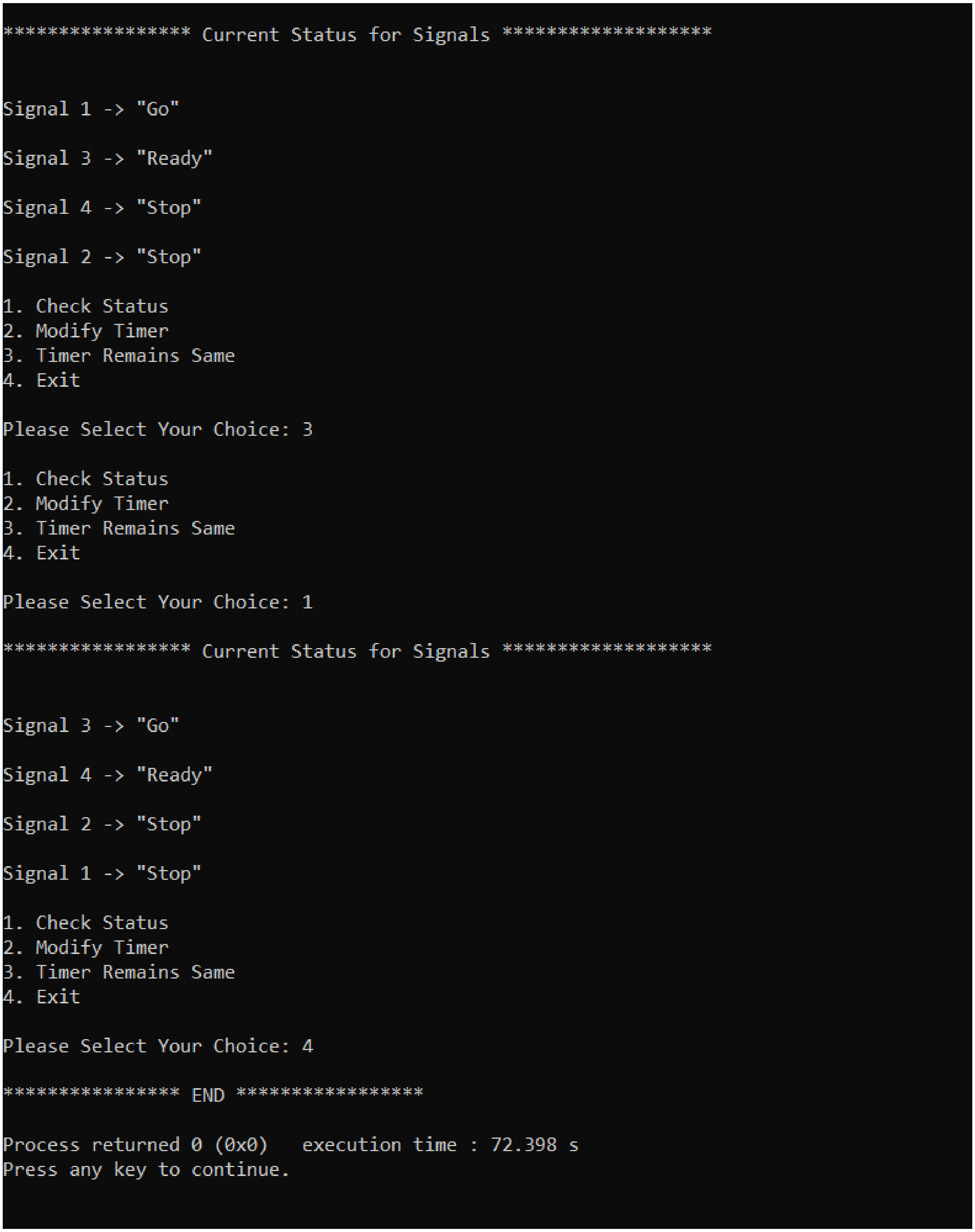
Screen Shots:



```
-------------------------------------------------------------------------------------------------------------------------------------------
                                                  WELCOME TO TRAFFIC SIGNAL SYSTEM
-------------------------------------------------------------------------------------------------------------------------------------------
                                                THE MOST EASIEST WAY TO HANDLE TRAFFIC
-------------------------------------------------------------------------------------------------------------------------------------------



************************ Enter the no. of Vehicles for Specific Signals to Work Accordingly ****************************

Signal 1: 45

Signal 2: 74

Signal 3: 12

Signal 4: 89


********************* Sorted Traffic ****************

 Signal  Vehicles
 4  ->   89
 2  ->   74
 1  ->   45
 3  ->   12

******************* Initial Scenario ***************


Signal 4 -> "Go"

Signal 2 -> "Ready"

Signal 1 -> "Stop"

Signal 3 -> "Stop"

Please Set Your Initial Timer: 5

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 1

***************** Current Status for Signals *******************
```



```
Please Select Your Choice: 1

***************** Current Status for Signals *******************


Signal 2 -> "Go"

Signal 1 -> "Ready"

Signal 3 -> "Stop"

Signal 4 -> "Stop"

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 11

Invalid Choice
1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 2

Please Enter Your Modified Timer: 7

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 1

***************** Current Status for Signals *******************


Signal 1 -> "Go"

Signal 3 -> "Ready"

Signal 4 -> "Stop"

Signal 2 -> "Stop"

1. Check Status
2. Modify Timer
```



```
***************** Current Status for Signals *******************


Signal 1 -> "Go"

Signal 3 -> "Ready"

Signal 4 -> "Stop"

Signal 2 -> "Stop"

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 3

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 1

***************** Current Status for Signals *******************


Signal 3 -> "Go"

Signal 4 -> "Ready"

Signal 2 -> "Stop"

Signal 1 -> "Stop"

1. Check Status
2. Modify Timer
3. Timer Remains Same
4. Exit

Please Select Your Choice: 4

**************** END *****************

Process returned 0 (0x0)   execution time : 72.398 s
Press any key to continue.
```