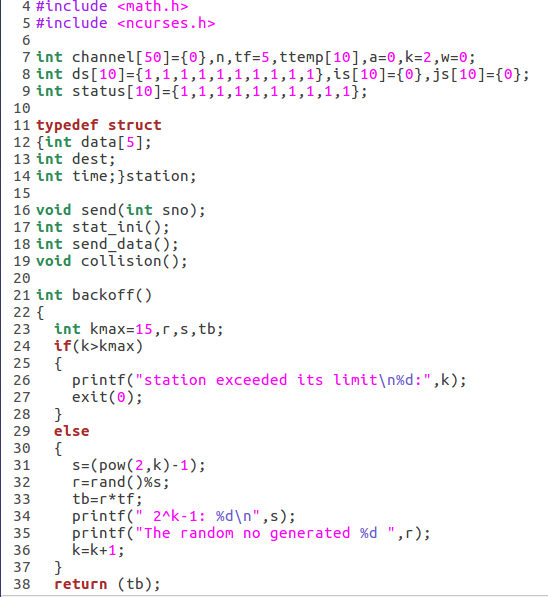
**Assignment 3: Implement p-persistent CSMA and CSMA/CD.**

(Carrier-Sense Multiple Access with Collision Detection)

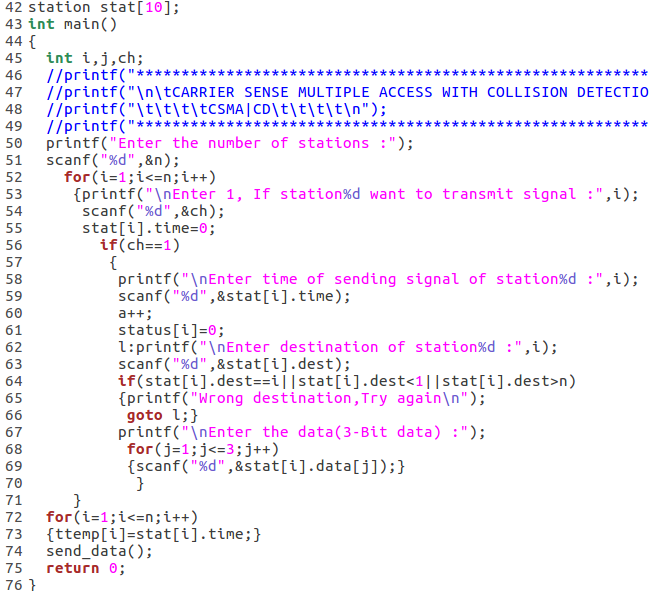
**1) What is CSMA/CD?**

CSMA/CD is a media access control method. It uses carrier-sensing to defer transmissions until no other stations are transmitting. This is used in combination with collision detection in which a transmitting station detects collisions by sensing transmissions from other stations while it is transmitting a [frame](https://en.wikipedia.org/wiki/Frame_(telecommunications)). When this collision condition is detected, the station stops transmitting that frame, transmits a jam signal, and then waits for a random time interval which is a number between 0 to 2^k-1, before trying to resend the frame.

A channel array is taken and is initailised to 50.

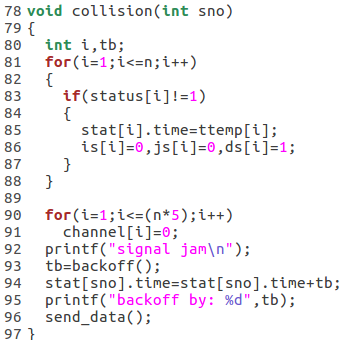
A structure station is declared which has an array containing the data, and integer denoting the destination and the time.

The function backoff returns the time for which the station will back off once a collision has happened. It has a variable k and a random number is generated between 0 and 2^k-1 for which the station will back off.

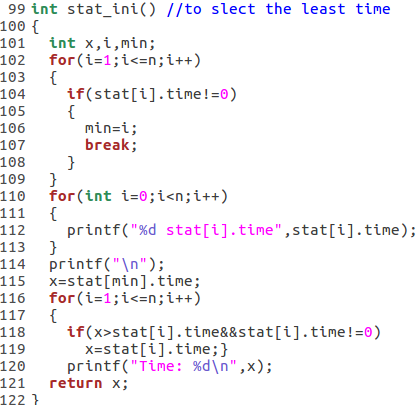


A channel is taken which can have a maximum of 10 stations.

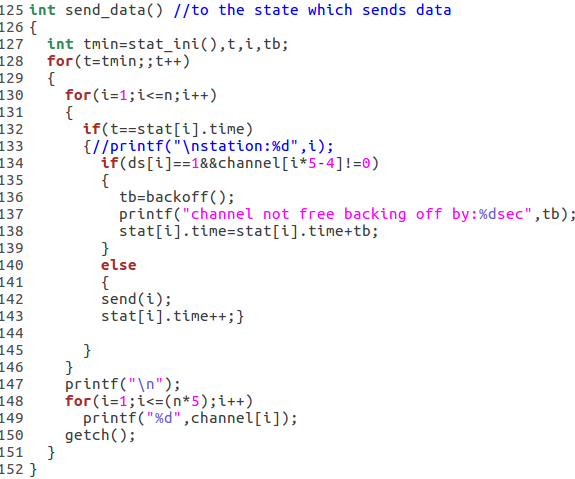
In the main function the input of the number of staions is taken and for each corresponding station the time, destionation and data is also taken. The send\_data function is called for each stations that wants to send.



Function to determine if collision has occurred for a station number. If the collision has occurred signal jam will be printed and it will back off and will try to send the data again.



This function selects the station with the least time which wants to send data. The minimum time is returned.



This function uses the minimum time to find out which station has to send the data. If the channel is not free it waits by the back off time and the send function is called again but with increasing value of the time.

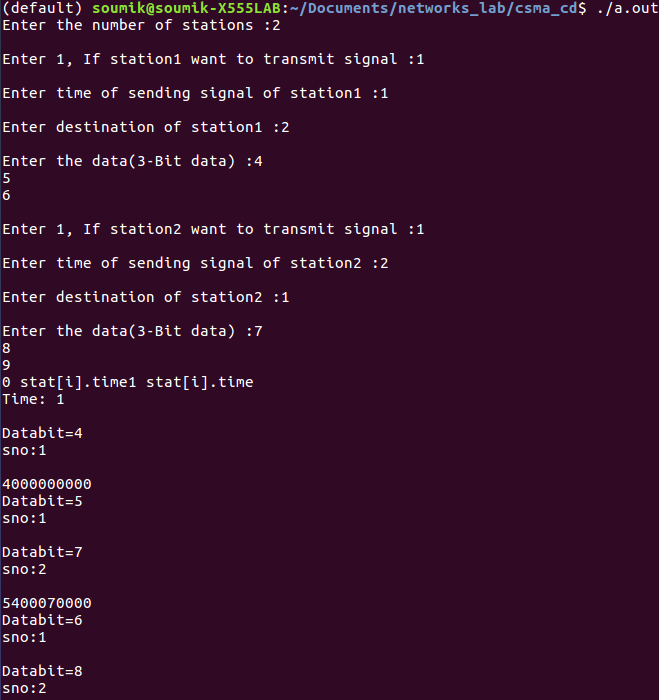


The send function is actually responsible for sending the data of the station which is given as an argument. The channel was initialised to 0. If for a particular bit the channel is not 0 that means it already in use by some other station then collision has occurred. This is the part where collision is detected

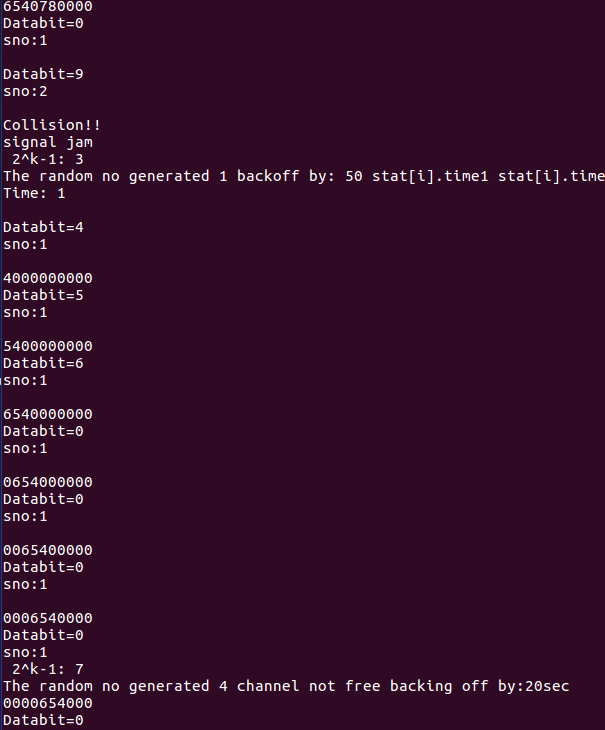


If the collision has not occurred the data bit in the channel will be set to the data. If the status of the stations are same then data sending is successful. When all the stations have sent their data all signals sent successfully will be printed.

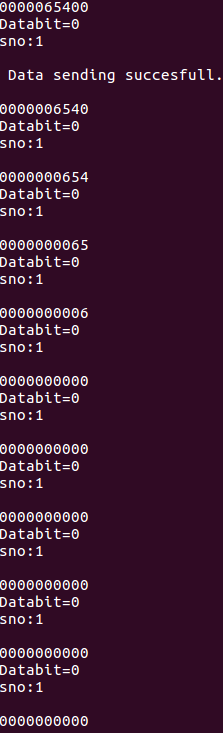
Output:



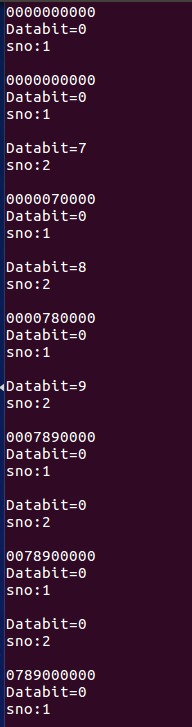
The main takes the input of number of stations, time,data and destination of each station. Since 2 stations are there, 10 is the size of the channel. At time t=1 only the station 1 sends data. So its just 4. At time=2 both station 1 and 2 send data. So output of channel is 540007000.



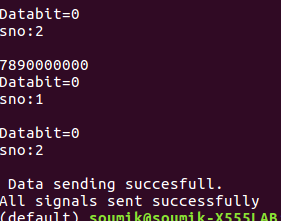
At time =4, both the stations want to change the 4th bit data from 0. So there is a collision. Therefore the backoff function will be called and a random number for which it will wait is generated. The data is sent again.



The data from the station 1 is sent first. When all the bits are sent data sending successful is printed. Next the station 2waits for the time as specified by the backoff function.



After the backoff time the data of station 2 will be sent again. This time there wont be any collision as the station 1 has already sent the data.



After all the data of the station 2 is sent All signals sent will be printed.