(i)Bit-Map

#include<bits/stdc++.h>

using namespace std;

void contention(vector<int> &senders)

{

for(int i = 0; i <senders.size(); i++)

{

senders[i] = rand() % 2;

}

}

void displayInterestedSenders(vector<int> senders)

{

cout<<"\nThe stations having frame to transmit are: ";

for(int i = 0; i <senders.size(); i++)

{

if(senders[i] == 1)

{

cout<<i + 1<<" ";

}

}

cout<<endl;

}

void sendFrames(vector<int> senders)

{

for(int i = 0; i <senders.size(); i++)

{

if(senders[i] == 1)

{

cout<<"Station "<<i + 1<<" is sending frame\n";

}

}

}

int main()

{

srand(time(NULL));

int n;

cout<<" Please Enter the number of stations: ";

cin>>n;

int t;

cout<<"Please Enter the contention periods: ";

cin>>t;

vector<int> senders(n);

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

{

contention(senders);

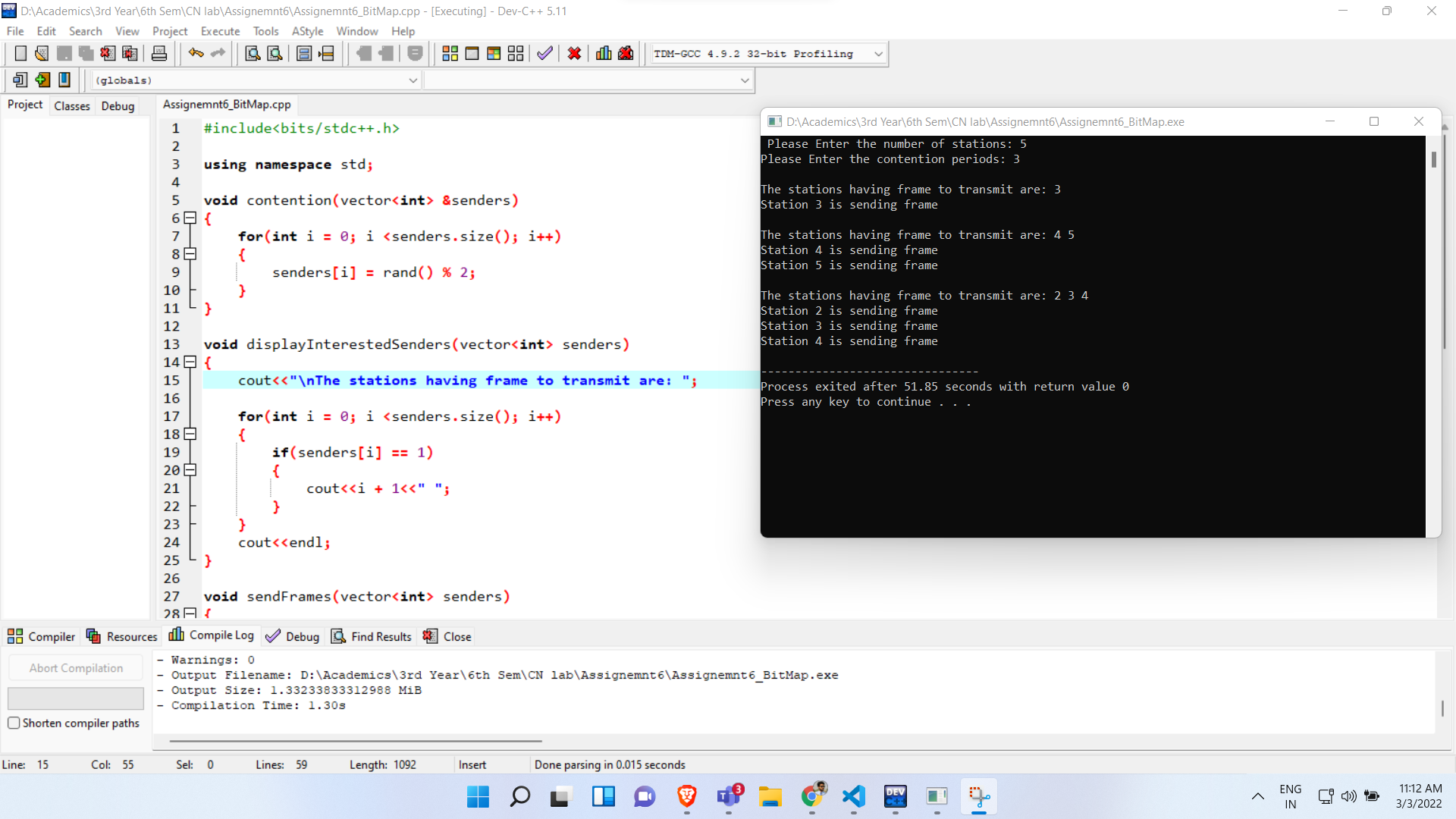
displayInterestedSenders(senders);

sendFrames(senders);

}

return 0;

}



**(ii)Token Bus**

#include<bits/stdc++.h>

using namespace std;

struct station{

int addr;

station \*nextAddr, \*prevAddr;

station()

{

addr = rand() % 100 + 1;

}

};

station\* formLogicalRing(vector<station> &s)

{

station \*head = &s[0];

for(int i = 0; i < s.size(); i++)

{

int nextInd, prevInd;

nextInd = (i + 1) % s.size();

prevInd = (s.size() + i - 1) % s.size();

s[i].nextAddr = &s[nextInd];

s[i].prevAddr = &s[prevInd];

}

return head;

}

void sendFrame()

{

int i = rand() % 2;

if(i == 0)

{

cout<<"No frame to send\n";

}

else

{

cout<<"Sending frame\n";

}

}

int main()

{

srand(time(NULL));

int n;

cout<<"Please!Enter the number of stations: ";

cin>>n;

int t;

cout<<"Please! Enter the number of cycles: ";

cin>>t;

vector<station> s(n);

station \*head = formLogicalRing(s);

cout<<"\nVirtual ring formed"<<endl;

station \*token = head;

cout<<"Token assigned to 1st station"<<endl;

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

{

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

{

cout<<"\nToken with station "<<token->addr<<endl;

sendFrame();

token = token->nextAddr;

}

}

return 0;

}

Output:

