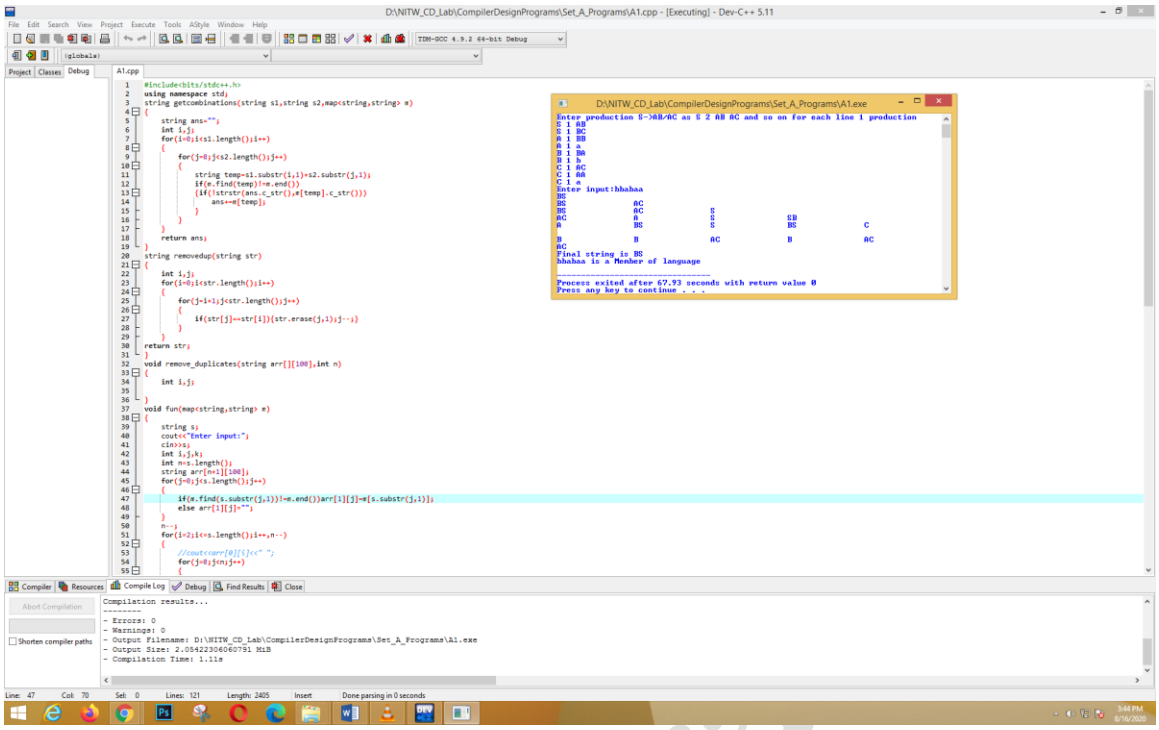
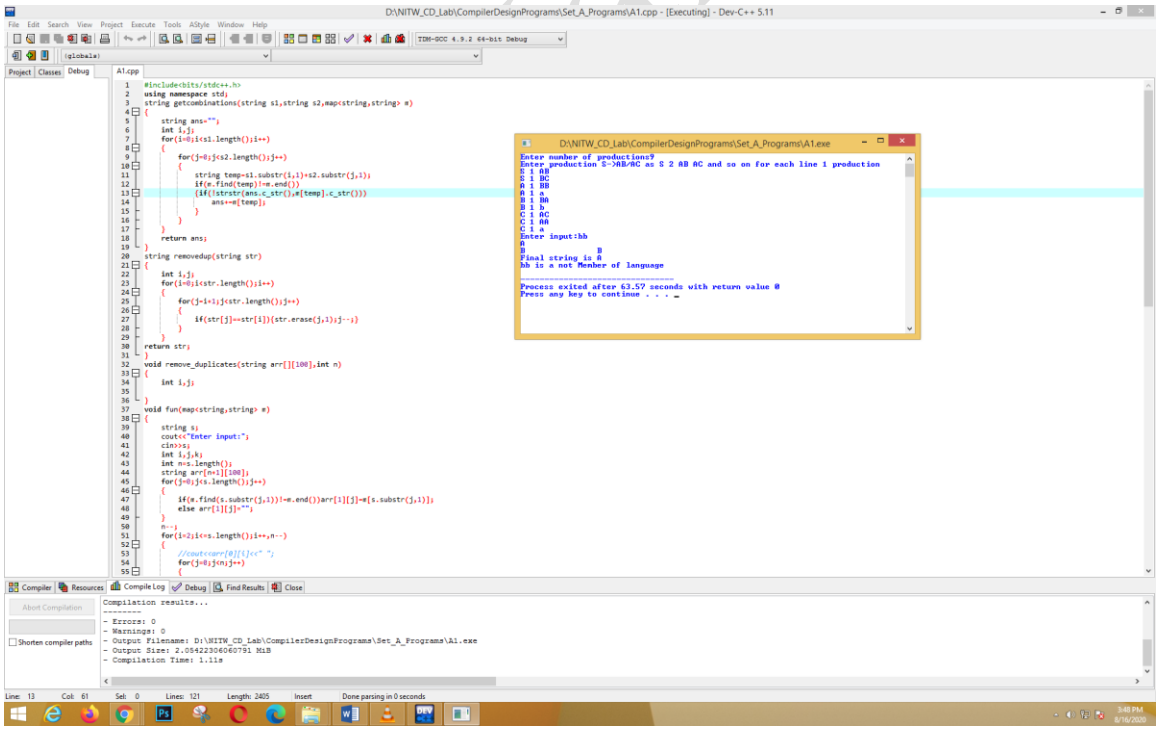
/* A1:Implementation of CYK Algorithm (you have studied in 'Theory of Computation') */

```
#include<bits/stdc++.h>
using namespace std;
/*returns concatenation of all the derivables(of size 2) formed
with
variables from string s1 and s2 whose entry exists in map m*/
string getcombinations(string s1,string s2,map<string,string> m)
{
      string ans="";
      int i,j;
      for(i=0;i<s1.length();i++)
      {
            for(j=0;j<s2.length();j++)
            {
                  string temp=s1.substr(i,1)+s2.substr(j,1); /*a
pair formed by picking up each variable from s1 and s2*/
                  if(m.find(temp)!=m.end())                  /*
checking if temp exists in the map*/
                  {if(!strstr(ans.c_str(),m[temp].c_str()))
                        ans+=m[temp];                        /*if
it's not already added,add it to ans*/
                  }
            }
      }
      return ans;
}
string removedup(string str)
{
      int i,j;
      for(i=0;i<str.length();i++)
      {
            for(j=i+1;j<str.length();j++)
            {
                  if(str[j]==str[i]){str.erase(j,1);j--;}
            }
      }
return str;
}
void remove_duplicates(string arr[][100],int n)
{
      int i,j;

}
/* fun f returns if the input word taken belongs to the language
or not */
void fun(map<string,string> m)
{
      string s; /*s stores the input string*/
```

```
      cout<<"Enter input:";
      cin>>s;
      int i,j,k;
      int n=s.length();
      string arr[n+1][100];
      for(j=0;j<s.length();j++)
      {   /*For each variable at position j in the string,store
it's corresponding map entry in arr[1][j]*/
            if(m.find(s.substr(j,1))!=m.end())
                arr[1][j]=m[s.substr(j,1)];
            else arr[1][j]="";
      }
      n--;
      /*for derivable of each size*/
      for(i=2;i<=s.length();i++,n--)
      {
            //cout<<arr[0][i]<<" ";
            for(j=0;j<n;j++)
            {
                  arr[i][j]="";
                  for(k=1;k<i;k++)
                  {
                        string s1=arr[k][j];      /*map entry
corresponding to derivable of size k and from position j*/
                        string s2=arr[i-k][j+k];  /*map entry
corresponding to derivable of size i-k and from position j+k*/
                        string temp=getcombinations(s1,s2,m);
/*gets all combo from s1,s2 whose entry exists in map m*/
                        if(!strstr(arr[i][j].c_str(),temp.c_str()))
                              arr[i][j]+=temp;                  /*if
it's not already added,add it to arr[i][j]*/
                  }
            }
      }
      n=s.length();

      for(i=1;i<=s.length();i++,n--)
      {
            for(j=0;j<n;j++)
            {
                  arr[i][j]=removedup(arr[i][j]); /*to remove
duplicated variables from arr[i][j]*/
            }
      }
      n=1;
      //cout<<n;
      /* output the arr[n][] table */
      for(i=s.length();i>0;i--,n++)
      {
            for(j=0;j<n;j++)
            {
                  cout<<arr[i][j]<<" ";
```

```
                for(k=0;k<15-arr[i][j].length();k++)cout<<" ";
            }
            cout<<"\n";
      }
      cout<<"Final string is "<<arr[s.length()][0]<<"\n";
      string final=arr[s.length()][0];
      string start="S";
      /*if S is not part of final string,then input string is not
a member of the language,otherwise yes*/
      if(strstr(final.c_str(),start.c_str()))cout<<s<<" is a
Member of language\n";
      else cout<<s<<" is a not Member of language\n";
      return;
}
int main()
{
      int n,m,i,j;
      map<string,string> mp;  /*mp stores concantenation of all
the headers corresponding to each derivable*/
      string header,s;
      cout<<"Enter number of productions";
      cin>>n;   /*n stores the number of productions*/
      cout<<"Enter production S->AB/AC as S 2 AB AC and so on for
each line 1 production\n";
      for(i=0;i<n;i++)
      {
            //cout<<"Enter header:";
            cin>>header;
            //cout<<"Enter number of derivables for "<<header<<"
:";
            cin>>m;
            for(j=0;j<m;j++)
            {
                  cin>>s;  //s=>the derivable
                  if(mp.find(s)!=mp.end())mp[s]=mp[s]+header;
                  else mp[s]=header;
            }
      }
      /*for(map<string,string>::iterator
it=mp.begin();it!=mp.end();it++)
      {
            cout<<it->first<<" "<<it->second<<"\n";
      }*/
      fun(mp);
      return 0;
}
```


D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.cpp - [Executing] - Dev-C++ 5.11
D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.exe
Enter production S->AB/AC as S 2 AB AC and so on for each line 1 production
S 1 AB
S 1 BC
A 1 BB
A 1 a
B 1 BA
B 1 b
C 1 AC
C 1 AA
C 1 a
Enter input:bbabaa
Final string is BS
bbabaa is a Member of language
Process exited after 67.93 seconds with return value 0
Press any key to continue . . .
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.exe
- Output Size: 2.05422306060791 MiB
- Compilation Time: 1.11s


D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.cpp - [Executing] - Dev-C++ 5.11
D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.exe
Enter number of productions?
Enter production S->AB/AC as S 2 AB AC and so on for each line 1 production
S 1 AB
S 1 BC
A 1 BB
A 1 a
B 1 BA
B 1 b
C 1 AC
C 1 AA
C 1 a
Enter input:bb
Final string is B
bb is a not Member of language
Process exited after 63.57 seconds with return value 0
Press any key to continue . . .
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: D:\NITW_CD_Lab\CompilerDesignPrograms\Set_A_Programs\A1.exe
- Output Size: 2.05422306060791 MiB
- Compilation Time: 1.11s