```
C:\Users\sahil\OneDrive\Desk
How many number of productions ? :8
Enter productions Number 1 : E=TD
Enter productions Number 2 : D=$
Enter productions Number 3 : F=a
Enter productions Number 4 : F=(E)
Enter productions Number 5 : S=*E
Enter productions Number 6 : T=fs
Enter productions Number 7 : D=+TD
Enter productions Number 8 : S=$

 Find the FIRST of  :E

 FIRST(E)= {  f }
press 'y' to continue : Y

 Find the FIRST of  :D

 FIRST(D)= {  $  + }
press 'y' to continue : Y

 Find the FIRST of  :F

 FIRST(F)= {  a  ( }
press 'y' to continue : Y

 Find the FIRST of  :S

 FIRST(S)= {  *  $ }
press 'y' to continue : Y

 Find the FIRST of  :
T

 FIRST(T)= {  f }
press 'y' to continue : Y
```

C:\Users\sahil\OneDrive\Desktop\pr5.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

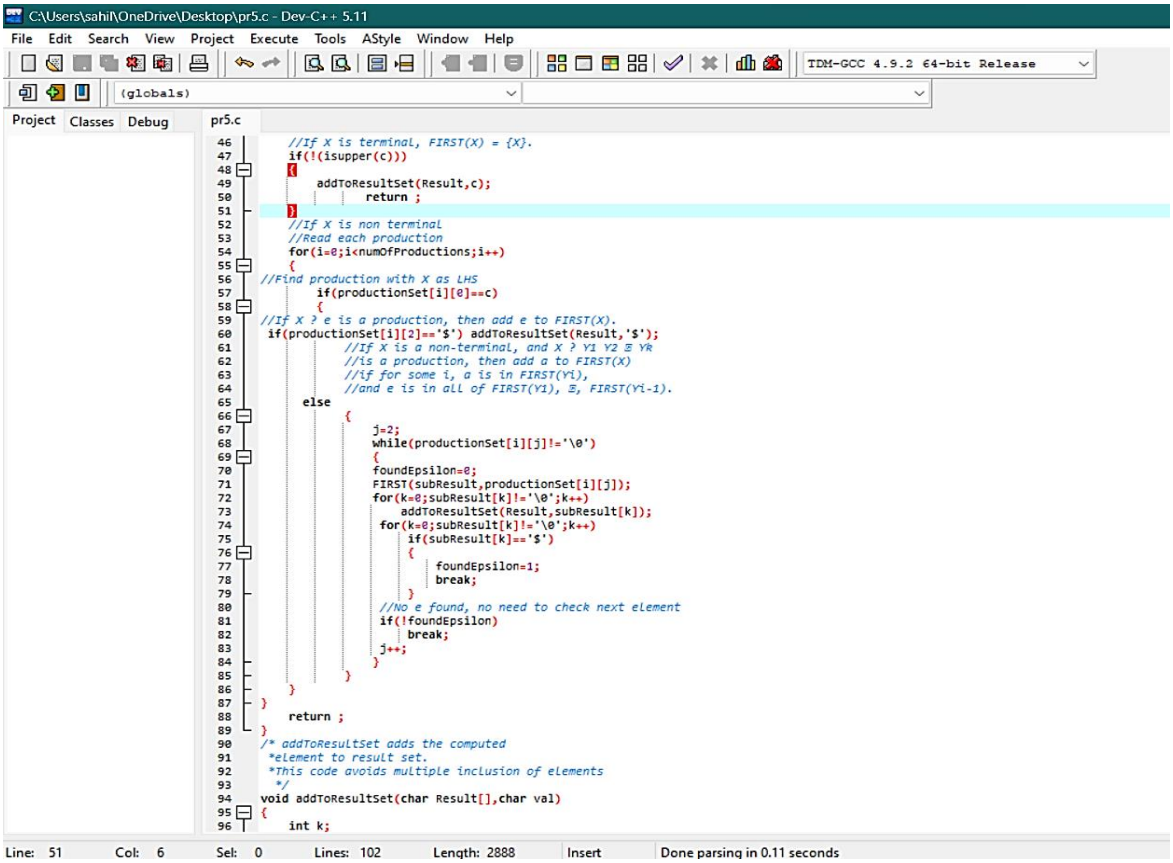
(globals)

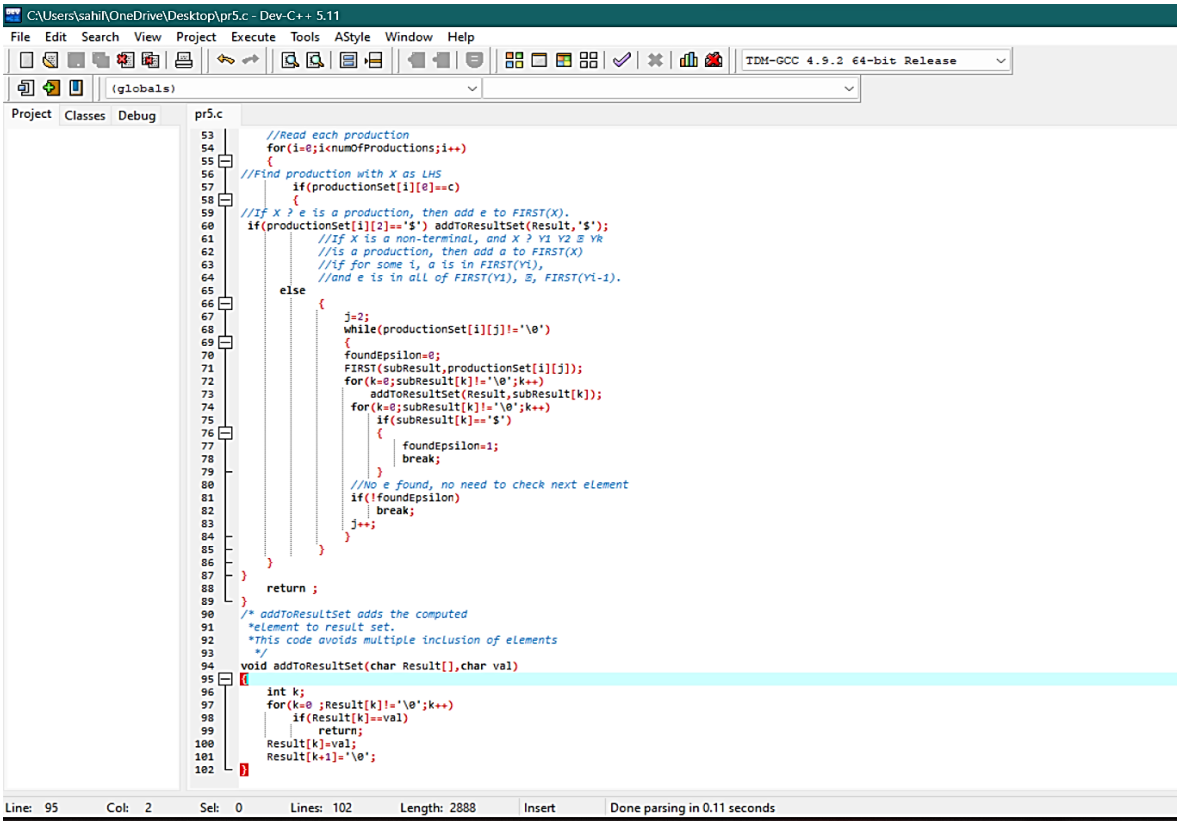
Project Classes Debug

pr5.c

```
1   #include<stdio.h>
2   #include<ctype.h>
3   void FIRST(char[],char );
4   void addToResultSet(char[],char);
5   int numOfProductions;
6   char productionSet[10][10];
7   main()
8   {
9       int i;
10      char choice;
11      char c;
12      char result[20];
13      printf("How many number of productions ? :");
14      scanf(" %d",&numOfProductions);
15      for(i=0;i<numOfProductions;i++)//read production string eg: E=E+T
16      {
17          printf("Enter productions Number %d : ",i+1);
18          scanf(" %s",productionSet[i]);
19      }
20      do
21      {
22          printf("\n Find the FIRST of  :");
23          scanf(" %c",&c);
24          FIRST(result,c); //Compute FIRST; Get Answer in 'result' array
25          printf("\n FIRST(%c)= { ",c);
26          for(i=0;result[i]!='\0';i++)
27          printf(" %c ",result[i]);       //Display result
28          printf("}\n");
29          printf("press 'y' to continue : ");
30          scanf(" %c",&choice);
31      }
32      while(choice=='y'||choice =='Y');
33  }
34  /*
35   *Function FIRST:
36   *Compute the elements in FIRST(c) and write them
37   *in Result Array.
38   */
39  void FIRST(char* Result,char c)
40  {
41      int i,j,k;
42      char subResult[20];
43      int foundEpsilon;
44      subResult[0]='\0';
45      Result[0]='\0';
46      //If X is terminal, FIRST(X) = {X}.
47      if(!(isupper(c)))
48      {
49          addToResultSet(Result,c);
50              return ;
51      }
```

Line: 2 Col: 18 Sel: 0 Lines: 102 Length: 2888 Insert Done parsing in 0.11 seconds


C:\Users\sahil\OneDrive\Desktop\pr5.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
pr5.c
//If X is terminal, FIRST(X) = {X}.
if(!(isupper(c)))
{
addToResultSet(Result,c);
return ;
}
//If X is non terminal
//Read each production
for(i=0;i<numOfProductions;i++)
{
//Find production with X as LHS
if(productionSet[i][0]==c)
{
//If X ? e is a production, then add e to FIRST(X).
if(productionSet[i][2]=='$') addToResultSet(Result,'$');
//If X is a non-terminal, and X ? Y1 Y2 ? Yk
//is a production, then add a to FIRST(X)
//if for some i, a is in FIRST(Yi),
//and e is in all of FIRST(Y1), ?, FIRST(Yi-1).
else
{
j=2;
while(productionSet[i][j]!='\0')
{
foundEpsilon=0;
FIRST(subResult,productionSet[i][j]);
for(k=0;subResult[k]!='\0';k++)
addToResultSet(Result,subResult[k]);
for(k=0;subResult[k]!='\0';k++)
if(subResult[k]=='$')
{
foundEpsilon=1;
break;
}
//No e found, no need to check next element
if(!foundEpsilon)
break;
j++;
}
}
}
}
return ;
}
/* addToResultSet adds the computed
*element to result set.
*This code avoids multiple inclusion of elements
*/
void addToResultSet(char Result[],char val)
{
int k;
Line: 51 Col: 6 Sel: 0 Lines: 102 Length: 2888 Insert Done parsing in 0.11 seconds


C:\Users\sahil\OneDrive\Desktop\pr5.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
pr5.c
//Read each production
for(i=0;i<numOfProductions;i++)
{
//Find production with X as LHS
if(productionSet[i][0]==c)
{
//If X ? e is a production, then add e to FIRST(X).
if(productionSet[i][2]=='$') addToResultSet(Result,'$');
//If X is a non-terminal, and X ? Y1 Y2 ? Yk
//is a production, then add a to FIRST(X)
//if for some i, a is in FIRST(Yi),
//and e is in all of FIRST(Y1), ?, FIRST(Yi-1).
else
{
j=2;
while(productionSet[i][j]!='\0')
{
foundEpsilon=0;
FIRST(subResult,productionSet[i][j]);
for(k=0;subResult[k]!='\0';k++)
addToResultSet(Result,subResult[k]);
for(k=0;subResult[k]!='\0';k++)
if(subResult[k]=='$')
{
foundEpsilon=1;
break;
}
//No e found, no need to check next element
if(!foundEpsilon)
break;
j++;
}
}
}
}
return ;
}
/* addToResultSet adds the computed
*element to result set.
*This code avoids multiple inclusion of elements
*/
void addToResultSet(char Result[],char val)
{
int k;
for(k=0 ;Result[k]!='\0';k++)
if(Result[k]==val)
return;
Result[k]=val;
Result[k+1]='\0';
}
Line: 95 Col: 2 Sel: 0 Lines: 102 Length: 2888 Insert Done parsing in 0.11 seconds