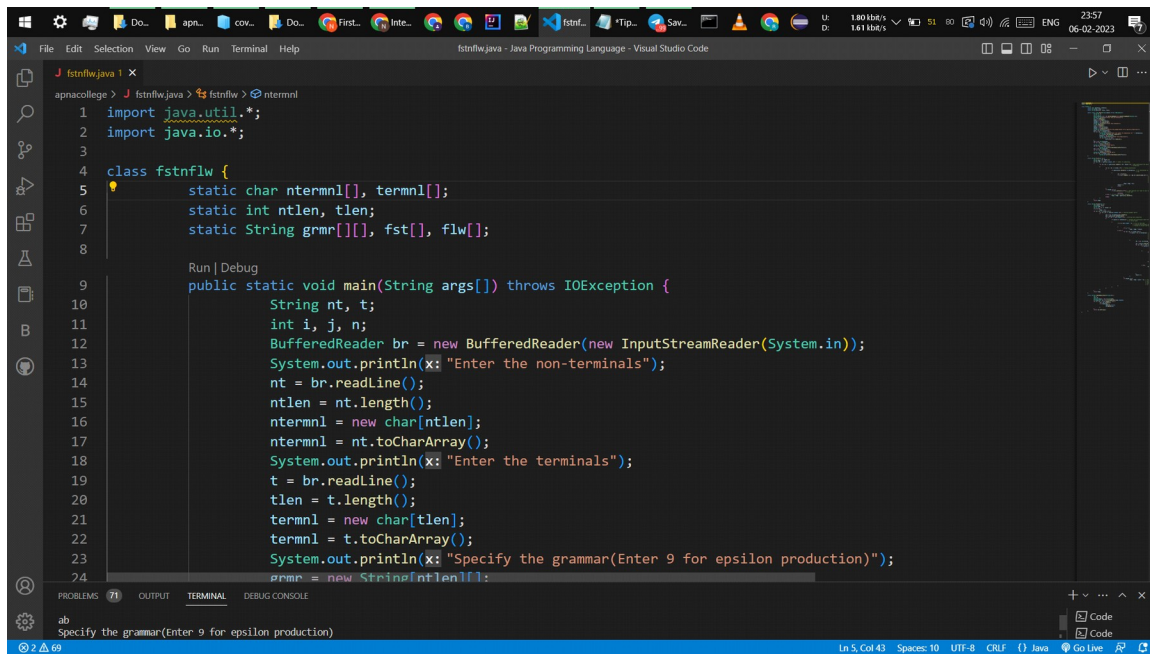
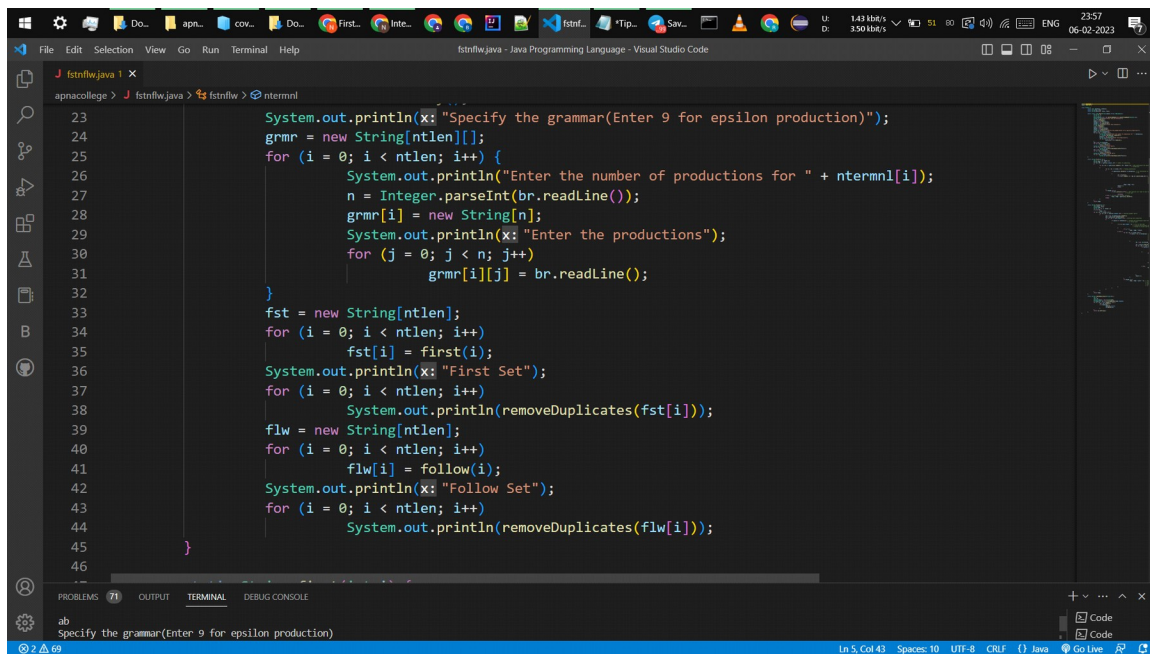


CODE :



```
1 import java.util.*;
2 import java.io.*;
3
4 class fstnflw {
5     static char ntermnl[], termnl[];
6     static int ntlen, tlen;
7     static String grmr[][], fst[], flw[];
8
9     Run | Debug
10    public static void main(String args[]) throws IOException {
11        String nt, t;
12        int i, j, n;
13        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
14        System.out.println(x: "Enter the non-terminals");
15        nt = br.readLine();
16        ntlen = nt.length();
17        ntermnl = new char[ntlen];
18        ntermnl = nt.toCharArray();
19        System.out.println(x: "Enter the terminals");
20        t = br.readLine();
21        tlen = t.length();
22        termnl = new char[tlen];
23        termnl = t.toCharArray();
24        System.out.println(x: "Specify the grammar(Enter 9 for epsilon production)");
25        grmr = new String[ntlen][];
```



```
23    System.out.println(x: "Specify the grammar(Enter 9 for epsilon production)");
24    grmr = new String[ntlen][];
25    for (i = 0; i < ntlen; i++) {
26        System.out.println("Enter the number of productions for " + ntermnl[i]);
27        n = Integer.parseInt(br.readLine());
28        grmr[i] = new String[n];
29        System.out.println(x: "Enter the productions");
30        for (j = 0; j < n; j++)
31            grmr[i][j] = br.readLine();
32    }
33    fst = new String[ntlen];
34    for (i = 0; i < ntlen; i++)
35        fst[i] = first(i);
36    System.out.println(x: "First Set");
37    for (i = 0; i < ntlen; i++)
38        System.out.println(removeDuplicates(fst[i]));
39    flw = new String[ntlen];
40    for (i = 0; i < ntlen; i++)
41        flw[i] = follow(i);
42    System.out.println(x: "Follow Set");
43    for (i = 0; i < ntlen; i++)
44        System.out.println(removeDuplicates(flw[i]));
45    }
46 }
```

```
File Edit Selection View Go Run Terminal Help
fstrflw.java - Java Programming Language - Visual Studio Code

J fstrflw.java X
apracollege > J fstrflw.java > fstrflw > nterminal
46
47 static String first(int i) {
48     int j, k, l = 0, found = 0;
49     String temp = "", str = "";
50     for (j = 0; j < grmr[i].length; j++) // number of productions
51     {
52         for (k = 0; k < grmr[i][j].length(); k++, found = 0) // when nonterminal has epsilon
53             // production
54         {
55             for (l = 0; l < ntlen; l++) // finding nonterminal
56             {
57                 if (grmr[i][j].charAt(k) == ntermnl[l]) // for nonterminal in first
58                     // set
59                 {
60                     str = first(l);
61                     if (!(str.length() == 1 && str.charAt(index: 0) == '9')) // when
62                         // epsilon
63                         // production
64                         // is the
65                         // only
66                         // nonterminal
67                         // production
68                     temp = temp + str;
69                     found = 1;
70                     break;
71                 }
72             }
73         }
74     }
75     return temp;
76 }
77
78 static String follow(int i) {
79     char pro[], chr[];
80     String temp = "";
81     int j, k, l, m, n, found = 0;
82     if (i == 0)
83         temp = "$";
84     for (j = 0; j < ntlen; j++) {
85         for (k = 0; k < grmr[i].length; k++) // entering grammar matrix
86         {
87             if (grmr[i][j].charAt(k) == ntermnl[l])
88             {
89                 if (l == 0)
90                 {
91                     if (str.contains("$")) // here epsilon will lead to next nonterminal's
92                         // first set
93                     continue;
94                 }
95                 else // if first set includes terminal
96                     temp = temp + grmr[i][j].charAt(k);
97             }
98         }
99     }
100     return temp;
101 }
```

PROBLEMS 71 OUTPUT TERMINAL DEBUG CONSOLE

ab Specify the grammar(Enter 9 for epsilon production)

Ln 5, Col 43 Spaces: 10 UTF-8 CRLF (J) Java Go Live

```
File Edit Selection View Go Run Terminal Help
fstrflw.java - Java Programming Language - Visual Studio Code

J fstrflw.java X
apracollege > J fstrflw.java > fstrflw > nterminal
68
69         temp = temp + str;
70         found = 1;
71         break;
72     }
73     if (found == 1) {
74         if (str.contains("$")) // here epsilon will lead to next nonterminal's
75             // first set
76         continue;
77     } else // if first set includes terminal
78         temp = temp + grmr[i][j].charAt(k);
79     break;
80 }
81 }
82 return temp;
83 }
84
85 static String follow(int i) {
86     char pro[], chr[];
87     String temp = "";
88     int j, k, l, m, n, found = 0;
89     if (i == 0)
90         temp = "$";
91     for (j = 0; j < ntlen; j++) {
92         for (k = 0; k < grmr[i].length; k++) // entering grammar matrix
93         {
94             if (grmr[i][j].charAt(k) == ntermnl[l])
95             {
96                 if (l == 0)
97                 {
98                     if (str.contains("$")) // here epsilon will lead to next nonterminal's
99                         // first set
100                     continue;
101                 }
102                 else // if first set includes terminal
103                     temp = temp + grmr[i][j].charAt(k);
104             }
105         }
106     }
107     return temp;
108 }
```

PROBLEMS 71 OUTPUT TERMINAL DEBUG CONSOLE

ab Specify the grammar(Enter 9 for epsilon production)

Ln 5, Col 43 Spaces: 10 UTF-8 CRLF (J) Java Go Live

```
91 for (j = 0; j < ntlent; j++) {
92     for (k = 0; k < grmr[j].length; k++) // entering grammar matrix
93     {
94         pro = new char[grmr[j][k].length()];
95         pro = grmr[j][k].toCharArray();
96         for (l = 0; l < pro.length; l++) // entering each production
97         {
98             if (pro[l] == ntermnl[i]) // finding the nonterminal whose follow set
99                 // is to be found
100             {
101                 if (l == pro.length - 1) // if it is the last
102                     // terminal/non-terminal then
103                     // follow of current non-terminal
104                 {
105                     if (j < i)
106                         temp = temp + flw[j];
107                 } else {
108                     for (m = 0; m < ntlent; m++) {
109                         if (pro[l + 1] == ntermnl[m]) // first
110                             // of next
111                             // non-terminal
112                             // otherwise
113                             // (else
114                             // later..)
```

```
114                                     // later..
115                                     {
116                                         chr = new char[fst[m]
117                                             .length()];
118                                         chr = fst[m].toCharArray();
119                                         for (n = 0; n < chr.length; n++) {
120                                             if (chr[n] == '9') // if
121                                                 // first
122                                                 // includes
123                                                 // epsilon
124                                             {
125                                                 if (l + 1 == pro.length
126                                                     - 1)
127                                                     temp = temp + follow(
128                                                         j); // when
129                                                         // non-terminal
130                                                         // is
131                                                         // second
132                                                         // last
133                                             } else
134                                                 temp = temp + follow(
135                                                     m);
136                                         } else
137                                             temp = temp + chr[n]; // include
138                                             // whole
139                                             // first
```



```
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

else
    temp = temp + follow(
        m);
    } else
        temp = temp + chr[n]; // include
                                // whole
                                // first
                                // set
                                // except
                                // epsilon

        found = 1;
    }
}
if (found != 1)
    temp = temp + pro[l + 1]; // follow set
                                // will
                                // include
                                // terminal(else
                                // is here)

}
}
};
```

PROBLEMS 71 OUTPUT TERMINAL DEBUG CONSOLE

ab
Specify the grammar(Enter 9 for epsilon production)

Ln 5, Col 43 Spaces: 10 UTF-8 CRLF (Java) Go Live

```
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175

// is here)

}
}
return temp;
}

static String removeDuplicates(String str) {
    int i;
    char ch;
    boolean seen[] = new boolean[256];
    StringBuilder sb = new StringBuilder(seen.length);
    for (i = 0; i < str.length(); i++) {
        ch = str.charAt(i);
        if (!seen[ch]) {
            seen[ch] = true;
            sb.append(ch);
        }
    }
    return sb.toString();
}
```

PROBLEMS 71 OUTPUT TERMINAL DEBUG CONSOLE

ab
Specify the grammar(Enter 9 for epsilon production)

Ln 175, Col 2 Spaces: 10 UTF-8 CRLF (Java) Go Live

OUTPUT :

```
PS C:\Users\bhave\Desktop\Java Programming Language\apnacollege> cd "C:\Users\bhave\Desktop\Java Programming Language\apnacollege\" ; if ($?) { javac fstnflw.java } ; if ($?) { java fs
fstnflw }
Enter the non-terminals
SABC
Enter the terminals
ab
Specify the grammar(Enter 9 for epsilon production)
Enter the number of productions for S
1
Enter the productions
aABC
Enter the number of productions for A
2
Enter the productions
a
bb
Enter the number of productions for B
2
Enter the productions
a
9
Enter the number of productions for C
2
Enter the productions
b
9
First Set
a
ab
a9
b9
Follow Set
$
ab$
b$
$
PS C:\Users\bhave\Desktop\Java Programming Language\apnacollege>
```