

EXPERIMENT-8

AIM:

Write a C program to find FOLLOW() - predictive parser for the given grammar.

PROGRAM:

The screenshot shows the Code::Blocks IDE interface. The top menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar contains various icons for file operations. The main window displays the code for EXP_1.cpp. The code implements a predictive parser to find the FOLLOW set of a non-terminal symbol from a given production rule. It uses standard C headers like stdio.h, string.h, and ctype.h. The parser reads a production rule (e.g., A=aB) and a non-terminal symbol (e.g., B), then iterates through the grammar to determine which terminals follow the non-terminal B. The code uses character arrays and loops to achieve this. Below the code editor is a tab bar with Compiler, Resources, Compile Log, Debug, Find Results, and Close. The Compile Log tab is active, showing the compilation results. It indicates 0 errors and 0 warnings, and provides details about the output file (C:\Users\Haritha\OneDrive\Documents\EXP_1.exe), output size (129.4423828125 KiB), and compilation time (0.488s).

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 int main() {
5     char grammar[50], symbol;
6     int i, found = 0;
7     printf("Enter a production (Example: A=aB): ");
8     scanf("%s", grammar);
9     printf("Enter the non-terminal to find FOLLOW of: ");
10    scanf(" %c", &symbol);
11    printf("\nFOLLOW(%c) = { ", symbol);
12    int len = strlen(grammar);
13    for (i = 0; i < len; i++) {
14        if (grammar[i] == symbol && i + 1 < len) {
15            if (!isupper(grammar[i + 1])) {
16                printf("%c ", grammar[i + 1]);
17                found = 1;
18            }
19        }
20    }
21    if (grammar[len - 1] == symbol) {
22        printf("$ ");
23        found = 1;
24    }
25    if (!found)
26        printf(" }");
27 }
```

OUTPUT:

The screenshot shows a terminal window with a black background and white text. The window title is C:\Users\Haritha\OneDrive\Documents. The command prompt shows the user entering the production rule A=aB and the non-terminal symbol B. The program then outputs the FOLLOW set of B, which is the empty set ({}). Finally, it prints a standard message indicating the process exited successfully after 37.19 seconds.

```
C:\Users\Haritha\OneDrive\Documents> Enter a production (Example: A=aB): A=aB
Enter the non-terminal to find FOLLOW of: B
FOLLOW(B) = { }

Process exited after 37.19 seconds with return value 0
Press any key to continue . . .
```