1. Design and Implementation of a Keyword Recognizer using Flex and Bison

Description: This project involves the design and implementation of a simple **keyword recognizer** using **Flex** and **Bison**. The program takes input from the user and checks whether the given word is a valid keyword such as `if`, `for`, `while`, or `else`. **Flex** is used to perform the lexical analysis and match these specific keywords, and **Bison** is used to define the grammar rules and handle valid keyword detection. When a valid keyword is entered, the compiler prints a confirmation message. This project demonstrates the basic concept of lexical token recognition and parsing in the context of programming language keywords, serving as an introductory step to compiler design.

Bison code:

```
%{
#include<stdio.h>
int yylex(void);
void yyerror(const char *s);
%}

%token KEYWORD

%%
start:
|start input
;
input:
KEYWORD {printf("valid keyword.");}
%%

void yyerror(const char *s)
{printf("syntax error: %s\n",s);}

int main()
{yyparse();
return 0;}
```

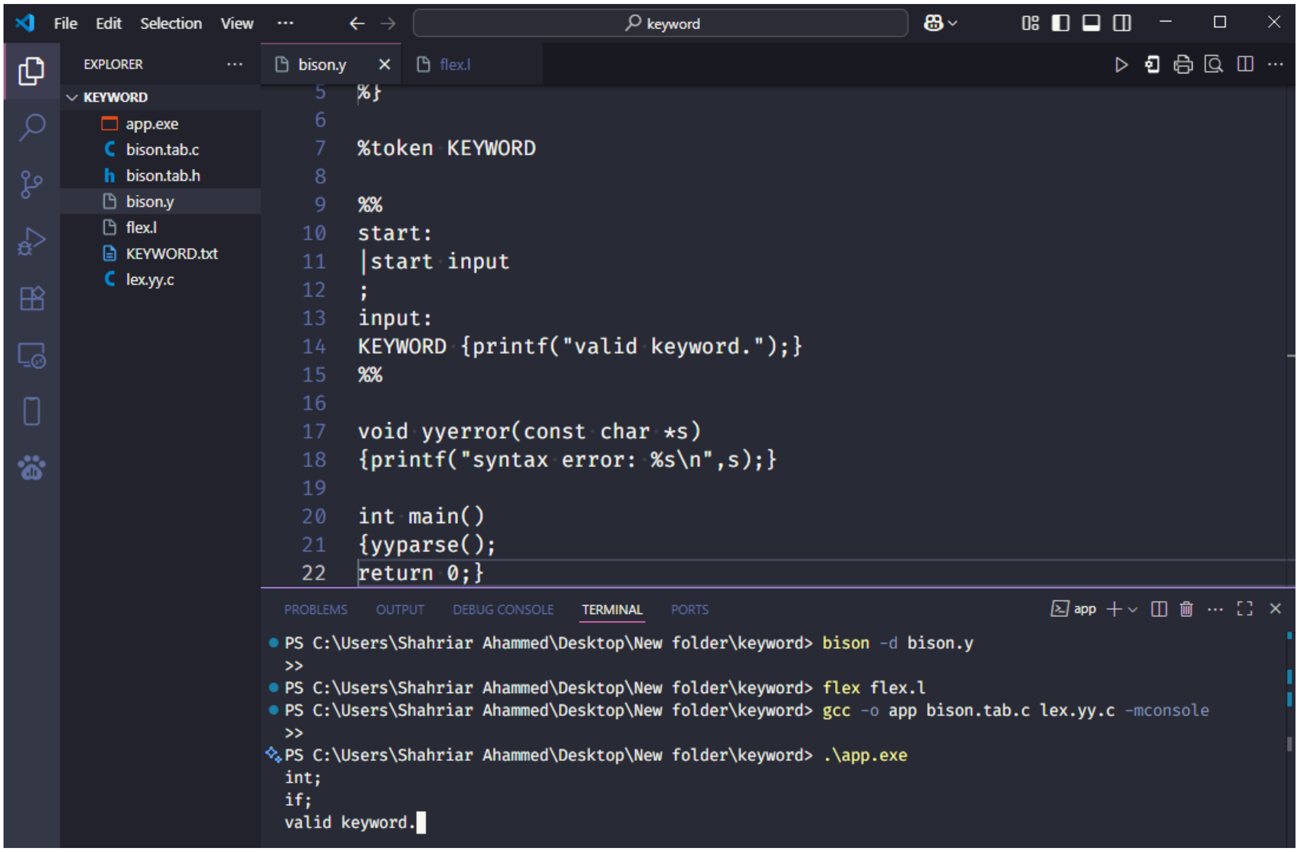
Flex code:

```
%{
#include "cal.tab.h"
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
%}

%%
[0-9]+ {yylval.num=atoi(yytext); return NUMBER;}
[ \n\t] ;
. {return yytext[0];}
%%

int yywrap()
{return 1;}
```

Output :



```
%}

%token KEYWORD

%%
start:
|start input
;
input:
KEYWORD {printf("valid keyword.");}
%%

void yyerror(const char *s)
{printf("syntax error: %s\n",s);}

int main()
{yyparse();
return 0;}
```

```
PS C:\Users\Shahriar Ahammed\Desktop\New folder\keyword> bison -d bison.y
>>
PS C:\Users\Shahriar Ahammed\Desktop\New folder\keyword> flex flex.l
PS C:\Users\Shahriar Ahammed\Desktop\New folder\keyword> gcc -o app bison.tab.c lex.yy.c -mconsole
>>
PS C:\Users\Shahriar Ahammed\Desktop\New folder\keyword> .\app.exe
int;
if;
valid keyword.
```