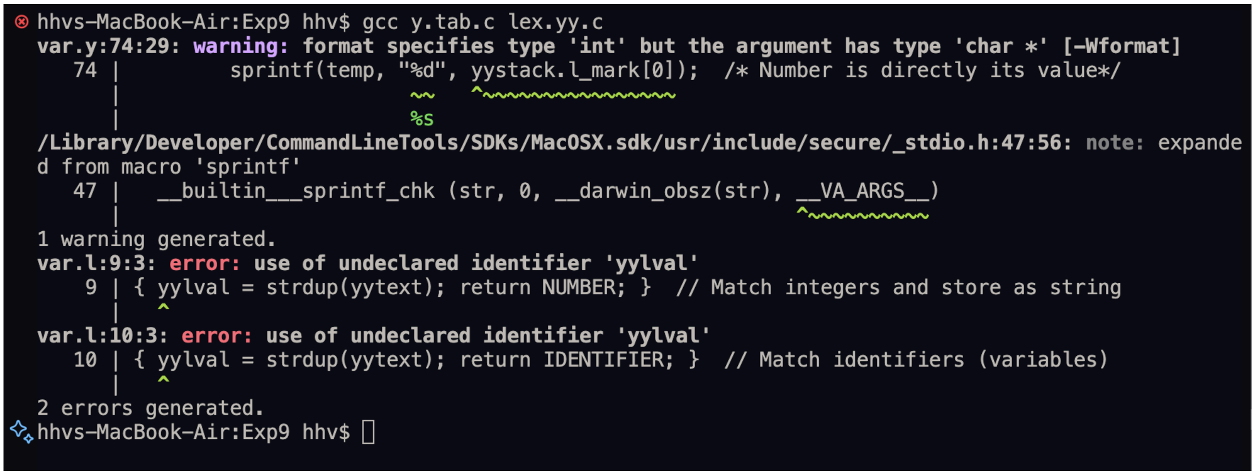
P HARIHARAVISWANATHAN 220701082

Exp8

Question: To design and implement a LEX and YACC program that generates three-address code (TAC) for a simple arithmetic expression or program. The program will: Recognize expressions like addition, subtraction, multiplication, and division. Generate three-address code that represents the operations in a way that could be directly translated into assembly code or intermediate code for a compiler.

Output:



```
⊗ hhvs-MacBook-Air:Exp9 hhv$ gcc y.tab.c lex.yy.c
var.y:74:29: warning: format specifies type 'int' but the argument has type 'char *' [-Wformat]
   74 |         sprintf(temp, "%d", yystack.l_mark[0]);  /* Number is directly its value*/
      |                       ~~    ^~~~~~~~~~~~~~~~~
      |                       %s
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/secure/_stdio.h:47:56: note: expande
d from macro 'sprintf'
   47 |   __builtin___sprintf_chk (str, 0, __darwin_obsz(str), __VA_ARGS__)
      |                                                        ^~~~~~~~~~~
1 warning generated.
var.l:9:3: error: use of undeclared identifier 'yylval'
    9 | { yylval = strdup(yytext); return NUMBER; }  // Match integers and store as string
      |   ^
var.l:10:3: error: use of undeclared identifier 'yylval'
   10 | { yylval = strdup(yytext); return IDENTIFIER; }  // Match identifiers (variables)
      |   ^
2 errors generated.
hhvs-MacBook-Air:Exp9 hhv$ 
```