| %{  #include <stdio.h>  #include <stdlib.h>  %}    %token NUMBER    %%  expression: expression '+' term { $$ = $1 + $3; }  | expression '-' term { $$ = $1 - $3; }  | term  ;    term: NUMBER { $$ = $1; }  ;    %%    int main(void) {  printf("Enter an arithmetic expression:\n");  yyparse();  return 0;  }    void yyerror(const char \*s) {  fprintf(stderr, "Error: %s\n", s);  } |
| --- |

| %{  #include <stdio.h>  #include <stdlib.h>  %}    %token NUMBER    %%  expression: expression '+' term { $$ = $1 + $3; }  | expression '-' term { $$ = $1 - $3; }  | '(' expression ')' { $$ = $2; }  | term  ;    term: NUMBER { $$ = $1; }  ;    %%    int main(void) {  printf("Enter an arithmetic expression with parentheses:\n");  yyparse();  return 0;  }    void yyerror(const char \*s) {  fprintf(stderr, "Error: %s\n", s);  } |
| --- |

| %{  #include <stdio.h>  #include <stdlib.h>  %}    %token COMMAND PARAMETER    %%  command: COMMAND { printf("Command: %s\n", yytext); }  | COMMAND PARAMETER { printf("Command: %s, Parameter: %s\n", $1, $2); }  ;    %%    int main(void) {  printf("Enter command:\n");  yyparse();  return 0;  }    void yyerror(const char \*s) {  fprintf(stderr, "Error: %s\n", s);  } |
| --- |

| %{  #include <stdio.h>  #include <stdlib.h>  %}    %token TAG TEXT    %%  document: element  ;    element: '<' TAG '>' TEXT '</' TAG '>' { printf("Tag: %s, Text: %s\n", $2, $4); }  | '<' TAG '>' element '</' TAG '>'  ;    %%    int main(void) {  printf("Enter XML-like text:\n");  yyparse();  return 0;  }    void yyerror(const char \*s) {  fprintf(stderr, "Error: %s\n", s);  } |
| --- |

| %{  #include <stdio.h>  #include <stdlib.h>  %}    %token SELECT FROM WHERE IDENTIFIER    %%  query: SELECT IDENTIFIER FROM IDENTIFIER WHERE IDENTIFIER {  printf("Select: %s, From: %s, Where: %s\n", $2, $4, $6);  }  ;    %%    int main(void) {  printf("Enter SQL query:\n");  yyparse();  return 0;  }    void yyerror(const char \*s) {  fprintf(stderr, "Error: %s\n", s);  } |
| --- |