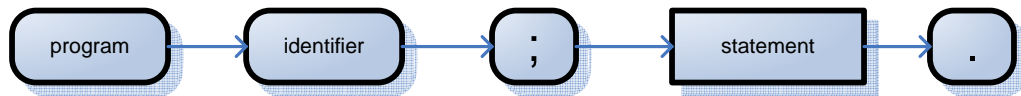
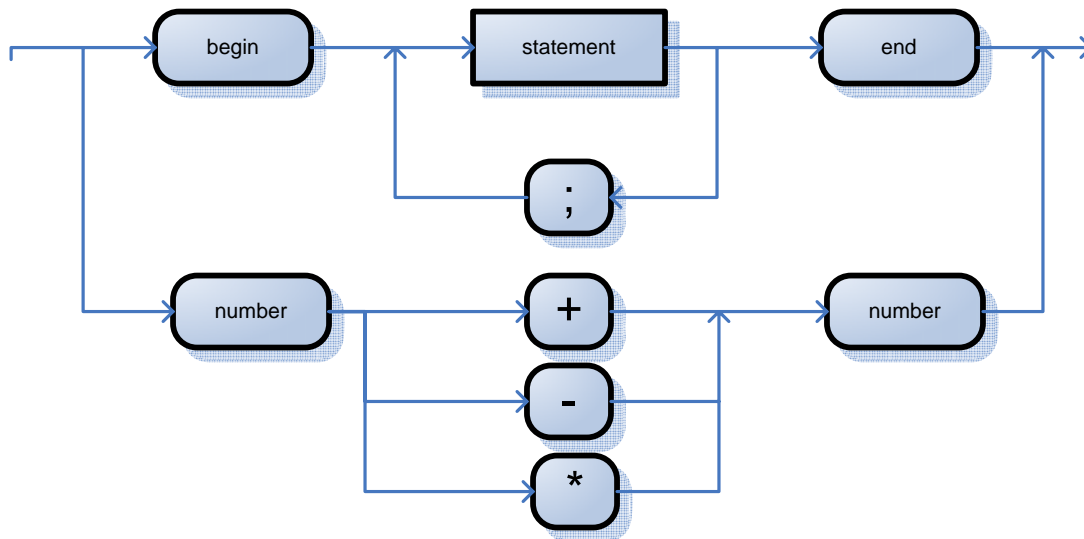


Buatlah parser untuk syntax graph seperti yang terdeskripsikan dibawah ini !

program :



statement :



Lampiran source code :

```
/*
    File name   : compiler.h
    Vers        : 0.01a
    Written by   : Setiadi Rachmat
    Date        : Fri Aug 28 10:08:03 WIT 2009
*/

#define isprogram      (token.attr == RWORD && token.value == PROGRAM)
#define isbegin        (token.attr == RWORD && token.value == BEGIN)
#define isend          (token.attr == RWORD && token.value == END)

#define isidentifier    (token.attr == IDENTIFIER)

#define isnumbr        (token.attr == NUMBER)

#define issemicolon    (token.attr == SYMBOL && token.value == SEMICOLON)
#define isperiod       (token.attr == SYMBOL && token.value == PERIOD)
```

```
/*
    File name   : compiler.c
    Vers        : 0.01a
    Written by   : Setiadi Rachmat
    Date         : Fri Aug 28 10:08:03 WIT 2009
*/

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include "scanner.h"
#include "compiler.h"

extern token_t token;

void program(void);
void statement(void);

int main(int argc, char *argv[])
{
    int c = 0;

    init_token();

    program();

    printf("Congratulations : No error is found\n");
}

void program(void)
{
    get_token();

    if (!isprogram) {
        printf("Error : Reserved Word 'program' is expected\n");
        exit(-1);
    }

    get_token();

    if (!isidentifier) {
        printf("Error : Identifier is expected after 'program' \n");
        exit(-1);
    }

    get_token();

    if (!issemicolon) {
        printf("Error : Symbol ';' is expected\n");
        exit(-1);
    }

    get_token();

    statement();

    if (!isperiod) {
        printf("Error : Symbol '.' is expected\n");
        exit(-1);
    }
}
```

```
}  
}  
  
void statement(void)  
{  
    /* silakan lengkapi */  
}
```

Testing Code :

| Input Source |
|--|
| Program example11; 526 * 321. |
| Program example12; Begin 100 + 21; 30 - 643; 10 * 32 End. |
| Program example13; Begin Begin 200 + 10; 30 - 8 End; 13 * 46 End. |