

تمرین 1 :

در فرآیند کامپایل ۴ مورد از مواردی که می توان با توجه به آنها کد را بهینه نمایید توضیح دهید؟

تمرین 2

با توجه به فایل `parser1.cpp` که کد آن در ادامه آمده است :

1. برنامه ای بنویسید که هر عدد اعشاری را دریافت و آن را تجزیه نماید.
2. برنامه ای بنویسید که هر عبارت شامل پرانتزهای تو در تو را دریافت و آن را تجزیه نماید.

تمرین 3

هر گروه با توجه به زبان برنامه نویسی خود اطلاعات زیر را در قالب یک فایل `word` ارسال می نماید.

1. تاریخچه ای از زبان برنامه نویسی مورد نظر.
2. نحوه فرآیند ترجمه در زبان مورد نظر.
3. محیط هایی که برنامه نویسی به آن زبان در آن امکان پذیر می باشد. و نحوه نصب و راه اندازی آن ها به صورت تصویری همراه با توضیحات کافی.
4. معرفی منابع مفید هم فارسی و هم زبان اصلی (منابع می تواند شامل سایت های اینترنتی و حتی ویدئو نیز باشد)

توجه : به روز بودن نکته مهمی در پاسخ گویی به سوالات بالا می باشد.

```
// It parses the string the user enters and also generates the parse tree.
// The Statement should end with a # symbol.
// A string in the language is : a+b#
// A string NOT in the language : a+-b#

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <conio.h>
char next;
void E(void);void T(void);
void S(void);void F(void);
void error(int);void scan(void);
void enter(char);void leave(char);void spaces(int);
int level = 0;

//The main should always be very simple
//First scan the string
//second check for end of string reached , if yes success and if not error.

//P ----> E '#'
int main(void){
    printf("Input:");
    scan(); E();
    if (next != '#') error(1);
    else printf("***** Successful parse *****\n");
    getch();
}

//E ----> T (('+'|'-') T)
void E(void){
    enter('E'); T();
    while (next == '+' || next == '-') {
        scan();
        T();}
    leave('E');}

//T ----> S (('*'|'/') S)
void T(void)
{
    enter('T'); S();
    while (next == '*' || next == '/') {
        scan(); S();
    }
    leave('T');
}

//S ----> F '^' S | F
void S(void)
{
```

```
enter('S'); F();
if (next == '^') {
    scan(); S();
}
leave('S');
}

//F ----> char | '(' E ')'
void F(void)
{
    enter('F');
    if (isalpha(next)) { scan();}
    else if (next == '(') {
        scan(); E();
        if (next == ')') scan();
        else error(2);
    }
    else { error(3);}
    leave('F');
}

//Scan the entire input
void scan(void){
    while (isspace(next = getchar()));
}

void error(int n)
{
    printf("\n*** ERROR: %i\n", n);
    getch();
    exit(1);
}

void enter(char name)
{
    spaces(level++);
    printf("+-%c\n", name);
}

void leave(char name)
{
    spaces(--level);
    printf("+-%c\n", name);
}

//TO display the parse tree
void spaces(int local_level)
{
    while (local_level-- > 0)
        printf("| ");
}
```